DOCUMENT RESUME

ED 262 760 IR 011 838

TITLE	[The Computing Teacher. Selected Articles on the use of Microcomputers in Language Arts Instruction.]
INSTITUTION	International Council for Computers in Education, Eugene, Oreg.
PUB DATE	85
NOTE	46p.
AVAILABLE FROM	International Council for Computers in Education, University of Oregon, 1787 Agate Street, Eugene, OR 97403.
PUB TYPE	Collected Works - General (020) -- Guides - Classroom Use - Materials (For Learner) (051) -- Viewpoints (120)
JOURNAL CIT	Computing Teacher; Dec 1983-Jun 1985
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	Computer Assisted Instruction; Computer Software; Elementary Secondary Education; Educational Material Evaluation; *Language Arts; Microcomputers; *North American Literature; Programing; Programing Languages; Teaching Methods; Word Processing
IDENTIFIERS	*Computer Uses in Education; Software Reviews

ABSTRACT

This document consists primarily of a compilation of nine articles, on the use of microcomputers in language arts instruction, that have been extracted from 1983-1985 issues of the journal "The Computing Teacher". Also included are various software reviews from the same period. The articles include: "Keyboarding Skills: Elementary, My Dear?" (Keith Wetzel/includes bibiliography); (2) "Keyboarding--A Must in Tomorrow's World" (Evelyn Kisner); (3) "Problems of Computer Assisted Instruction in Composition" (Robert M. Hertz); (4) "Teaching Literature with the Help of Microcomputers" (Elaine McNally Jarchow); (5) "The Electronic Writing Machine: Using Word Processors with Students" (Karen L. Piper); (6) "Computer Haiku: A Classroom Experiment with LOGO in the Language Arts" (Bates Hoffer and Pat Semmes); (7) "Follow 2--A Theme Commentary Program" (Leigh Howard Holmes); (8) "Bank Street Writer--Does the Tool Define the Process or Vice Versa?" (Randall Boone); and (9) "'But What Do I Write?'--Literary Analysis Made Easier" (Helen J. Schwartz/includes sample course syllabus). The second section of the document contains reviews of seven software programs: (1) KidWriter, (2) Story Maker, (3) Bank Street Story Book, (4) Story Tree, (5) Bank Street Writer (3 reviews), (6) Bank Street Speller, and (7) Quill. (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it
Minor changes have been made to improve reproduction quality
Points of view or opinions stated in this document do not necessarily represent official NIE position or policy

ED262760

Keyboarding Skills: Elementary, My Dear Teacher?

by
Keith Wetzel

I had high hopes for the computers-in-composition program I was evaluating last year. I imagined students eagerly writing more than ever before, taking the time to rework their compositions as they became more efficient in using the word processor. Although the eight to ten year olds and eight teachers worked diligently, the program floundered. Students neither revised more nor wrote more using a word processor than they did using paper and pencil. And not only that—the students seldom reached the revising stage. Part of the problem was that each student had only 30 minutes per week at the computer, an unrealistic amount of time to achieve the program goals. However, clearly a bigger factor was that students worked inefficiently at the keyboard.

Students found the letters they needed with difficulty, and in so doing the flow of their thoughts was interrupted. I watched student after frustrated student scan handwritten notes for the last word typed, and then look down for the key that represented the first letter of the next word, look up at the screen to verify the letter and correct letter position—and then lose the place. Fingers would point and eyes would search from paper to keyboard to screen and back to paper. The cycle continued and the frustration grew.

When I realized that the problem was inadequate keyboarding skills, I reviewed the literature and found that other educators had noted similar concerns: Students wrote faster by hand than by word processor and hated typing (Gottschalk in Brady, 1984); furthermore, once they had typed their essays, students had little time for revising (Loud in *CCN-Forum*, 1982). I found myself in general agreement. Although the third, fourth and fifth graders that I observed did not *hate* typing, they did enter text at a slower rate than their handwriting speed, and they often did not reach the revision stage of writing. The fact is, students who can't type have a hard time using a word processor.

What happens when students *can* type? Students who type better are more enthusiastic about using the computer for writing (Daiute et. al., 1981) and computer programming (Stoker, 1984). Students who have adequate keyboarding skills use their time at the computer efficiently—that is, they can concentrate on problem solving or composing, rather than on the mechanics of typing. ***Keyboarding is too important to leave to chance.*** For these reasons educators should make sure that keyboarding is taught and learned.

Keyboarding Issues

The computer keyboard exists to accept and to enter information. Computer keyboarding skills are primarily typing skills, i.e., striking desired keys accurately and quickly, including the skills needed to correctly use special keys such as ESC and CTRL.

Educators wishing to make sound decisions about keyboarding curricula must address several questions:

1. What criterion ought to be set for keyboarding competence at the elementary school level?

Some authors suggest that students ought to type about 25 gross words per minute (gwpm) before using a computer application program (Kisner, 1984; Minnesota Curriculum Services Center, 1984). However, I believe students need varying degrees of skill (this will be described in more detail later); 25 gwpm is not a realistic entry-level goal. (Gwpm is figured by dividing total keystrokes per minute by five, with no adjustment made for errors.) Students need to type quickly and accurately enough to make relatively efficient use of the computer and to accomplish the purposes of the academic program.

"I watched student after frustrated student scan handwritten notes ... and then look down for the key

... up at the screen

... and then lose the place."

© Copyright 1985 ICCE

IR011838

BEST COPY AVAILABLE

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
David Morsund
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Most fourth through sixth graders can copy from seven to 10 gross words per minute by hand, depending on the grade level (Groff, reported in Graham and Miller, 1980). Consequently the criterion for adequate keyboarding skills should at least equal that suggested by the handwriting norms. I observed 26 third, fourth and fifth graders using word processors with no touch-typing instruction. The average rate of transfer of text from paper to screen was 2.5 gwpm. This included a few students with some keyboarding skill. Those few who could type 7-10 gwpm did not scan the keyboard to locate each key, did not become frustrated by keyboard input, and did not lose their place on the screen or on the paper. I conclude that students who achieve 10 gwpm can make adequate use of the computer for tasks which require a significant amount of keyboard entry.

2. How much keyboarding is necessary?

The amount of time required for keyboarding instruction depends upon the computer application. Figure 1 illustrates my estimations of the keyboarding skill needed at the computer to accomplish various educational tasks.

The greater the amount of data entry, the higher the keyboarding skills required. Programs such as *Rocky's Boots* require no keyboarding skills—users work with a joystick, Mouse or KoalaPad. Drill and practice programs typically require students to use only a few keys; many times, for example, students select among options with a single keystroke. High-level keyboarding skills are required for entering computer programming code and for word processing.

Students acquire the "little" and "some" skill levels by learning how to move their fingers about the keyboard more quickly, starting from a base on the home row keys. To help students reach the "substantial" skill level, however, takes sustained time and effort.

3. How much time is needed for students to reach minimal proficiency in keyboarding?

Primary students can be taught the position of the home row keys and the correct finger for touching each key. This requires little additional instruction time. For example, when a reading lesson introduces a new letter, students can practice the letter on paper keyboards at their desks. If they can successfully use a computer for tasks that require a low level of keyboarding skill, they will be better prepared for keyboarding instruction when they enter grades three to five. This early preparation will become important to those teachers.

Degree of Keyboarding Skill Needed for Computer Applications

Application	Amount of Skill Required
• Logic Programs	
—*Rocky's Boots*	•
• Art Programs	
—*KoalaPad*	•
• Drill and Practice	•••••
• Computer Programming	
—Short programs	••••••••••••••
—Long programs	••••••••••••••••••••••••••••••••••••
• Word Processing	••••••••••••••••••••••••••••••••
	None Little Some Substantial

Figure 1.

To effectively use a computer application requiring a "substantial" degree of keyboarding skills requires a keyboarding curriculum. Teachers and students must commit time daily to accomplish proficiency in entering code for computer programming or text for word processing. ***Keyboarding instruction, however, cannot be integrated easily into the curriculum without replacing something;*** the question of time is critical.

"Keyboarding is too important to leave to chance."

Typing studies in the elementary school serve as examples of the time frame needed. McClurg (1984) found two studies which reported the typing speeds of intermediate students: Fifth and sixth graders typed an average of 40 wpm after one year of one-hour-per-day instruction (Ray, 1977), and fifth graders typed an average of 22 gwpm after nine weeks of instruction at 45 minutes a day (Kercher, 1984). Lexie Hendersen-Lancett (1984) reports on the time required for teaching elementary-age students keyboarding and word processing skills:

> "Children have shown they can learn both touch-typing and word processing in as little as eight hours! This enables them to create stories on the computer at speeds of 15 words per minute. This is quite adequate."

Such reports strengthen the case for the effectiveness of a minimal, but definite time allotment for keyboarding instruction.

Based on a review of the research, most students in grades three through five will average 10 gwpm after receiving teacher instruction and practice on a microcomputer typing tutorial for 35 minutes per day for four weeks. If this schedule is extended to nine weeks, the average student will type 15 to 20 gwpm. Computer programming, data base construction or word processing can begin when students achieve 10 gwpm, which also gives them something to work toward. Students' speed and accuracy should continue to improve as they use the computer for academic tasks, especially if the teacher follows up on previous instruction and practice.

I recommend a keyboarding program which will accomplish this minimal goal (10 gwpm) in a relatively short period of time. Elementary teachers are under many competing pressures to teach more subject areas in the course of a day. A keyboarding program has merit, but it does not deserve 50 minutes of daily instruction for a year! The benefits do warrant 35 minutes a day for four weeks in the intermediate grades. Not only will the investment be repaid through more efficient use of time at the keyboard, that minimum level of keyboarding skill is *essential* to some activities.

4. Who should teach keyboarding in the elementary school?

Some business education teachers argue that only persons such as business education majors or minors should teach keyboarding skills (Kisner, 1984; Holmquist, 1983; Alexander and Dickey-Olson in Schmidt, 1983). Clearly, persons trained to teach typing should do the teacher training, but I think regular classroom teachers should

© Copyright 1985 ICCE

BEST COPY AVAILABLE

instruct elementary students. Here the logistical problems and costs associated with using special teachers for keyboarding instruction must be balanced against the feasibility of such specialists training regular classroom teachers to do the instruction. Teachers who can type may be trained in as little as one day, and even those who do not type may be trained in as little as five weeks (Peggy Kennedy, 1984).

5. How should keyboarding skills be taught?

Keyboarding is usually taught by the touch-typing method. A common method of teaching keyboarding skills is the standard typing instruction offered in business education courses at the secondary level. Software tutorials such as *Typing Tutor* and *Microtyping*, and typing games such as *Mastertype* and *Type Attack* also offer keyboarding practice. Students use such software to learn to type on their own without benefit of formal instruction, although not necessarily with touch-typing techniques.

Researchers have examined microcomputer typing tutorial approaches which included only limited emphasis on touch-typing techniques. Schmidt and Stewart (1983) found that non-typing college students who received 10 hours of keyboarding practice on a microcomputer typing tutorial program achieved speed and accuracy rates commensurate with expectations for students in standard classes after 10 hours of instruction. Students in the study did comment on the inadequacy of the instruction regarding correct fingering and the too-quick introduction of new keys. The success of this approach may depend on the quality of the software used to teach or reinforce touch-typing skills.

Touch typing is important in the elementary school for several reasons. Beyond the beginner stage, touch typing methods are superior to two-finger hunt-and-peck methods (Stewart and Jones, 1983; Kisner, 1984). Gentner and Norman (1984) explain: "Touch-typing's major advantage comes when transcribing a manuscript: It eliminates the need to keep looking back and forth between the original and the keyboard." Business education teachers report that it is not easy to teach correct methods to students who have acquired improper typing habits (Stewart and Jones, 1983; Beverstock, 1984). Skills taught in the elementary school should be consistent with the skills students may subsequently need; for this reason I advocate the teaching of touch typing to elementary school students.

Program Considerations

Hardware

Few schools have a computer for each student. Beal and others (1983) found in a survey of schools in Washington state that only 36 percent of elementary schools had computers, and in those which had computers the student/computer ratio was 184 to one. In a nationwide survey, Henry Becker (1984) found that in elementary classrooms where students used computers, each student used one for an average of 20 minutes per week. If machine time is this limited, there is no time—or need—to teach keyboarding. However, many elementary students now have much more access to computers.

"Students with adequate keyboarding skills can use their time at the computer efficiently ... on problem solving or composing rather than on the mechanics of typing."

Time

If school officials are planning to place computers in the elementary school, or already have a sufficient number, a keyboarding program may be needed. How much computer time must each student have to accomplish a significant task? The critical amount of time needed to make a significant difference, according to David Moursund (1984), is 30 minutes per student per day. Classrooms approaching this level of computer use should include keyboarding curricula.

Keyboarding Curriculum

The specifications for a keyboarding curriculum have emerged from the discussion of the five questions discussed earlier. Before embarking on a keyboarding program, however, educators should ask these four threshold questions:

- Has an academic area been identified that needs to be improved through the use of computers?
- Can the identified problem best be solved by a computer solution?
- Are enough computers available for students to achieve the goals of the program?
- Does the computer application require frequent and repeated use of most of the keyboard?

If answers to the above questions are "yes," a keyboarding program is needed. Figure 1 shows the amount of skill needed for specific applications, with a minimal keyboarding goal for each student of 10 gwpm using touch-typing techniques as much as possible.

Principles of Keyboarding Instruction

Most texts and manuals agree on principles of keyboarding instruction. The Minnesota Curriculum Services Center (1984) indicates, for example, that:

1. Kinesthetic feedback in practice is important.
2. Watching fingers and keys in the early stages of learning is helpful.
3. Early emphasis should be on speed, not accuracy.
4. Knowledge of results helps develop skills.
5. Elementary students need to use several approaches in the early stages of instruction, e.g., teacher dictation of keys, games and written drills.

MECC '85

November 18-19, 1985
Bloomington, MN

For more information contact MECC, 3490 Lexington Avenue North, St. Paul, MN 55112.

ISECON '85
The Information Systems Education Conference

October 26-27, 1985
Houston, TX

For further information contact ISECON '85, Data Processing Management Association, 505 Busse Highway, Park Ridge, IL 60068-3191; ph. 312/825-8124.

© Copyright 1985 ICCE

Instructional Periods

Several sources suggest that instructional periods of 20 to 30 minutes are appropriate (Holmquist, 1983; Kisner, 1984; Minnesota Curriculum Services Center, 1984). After reviewing the research, I discussed the organization of a teaching period with keyboarding instructor Gale Daggett (1984), and observed her tutoring a student in keyboarding. I suggest this model:

Objective	Suggested Activity
A. Use correct keyboarding technique.	1.1 Students watch copy or screen rather than fingers or keyboard.
	1.2 Teacher(s) or volunteer(s) demonstrate posture, arm and wrist position, and finger positions for home keys according to software or text used.
B. Use the *A* and *S* keys, and space bar on computer keyboard.	2.1 Teacher(s) or volunteer(s) demonstrate and students practice the *A* keyfinger and space bar and then the *S* keyfinger and space bar according to text or software used.
	2.2 Students practice at desks as teacher monitors behaviors.
	2.3 Continue to practice tasks via software and a variety of drills on the computer.
	2.4 Practice real words as soon as possible.

Classroom or Computer Lab Organization

Elementary school computer(s) may be in the classroom or in a computer lab. Often the question is whether or not the computers should be divided equally among classrooms, where each classroom may have one or two computers—and perhaps just for part of a day. Both arrangements have advantages and disadvantages.

With a classroom arrangement:

1. The teacher presents a short lesson to the entire class on the skills to be practiced that day.
2. Students have a brief practice period at their desks.
3. Pairs of students practice at the computer throughout the day according to a schedule.

The advantages of a classroom arrangement are:

- Students do not need to leave the classroom to practice.
- The classroom teacher monitors student activities.
- One or two computers in a class can serve all the students.
- Nothing in the curriculum needs to be supplanted.

It should be noted that if students are regrouped out of their home classroom for much of the day, it is very difficult to schedule keyboarding practice.

The first two steps remain the same in the computer lab arrangement, but during step three all students practice at the same time. For example, if an elementary school has 13 computers in the lab and 26 students in the class, all students can practice in pairs at the same time. While one student works at the keyboard, the partner dictates words, monitors technique and records scores. Students trade positions after three minutes of practice, with each student practicing for five periods of three minutes each. Typing tutorial programs require intense concentration, and distributed practice makes sense in this instance. However, the very best practice schedule is open to investigation. The advantages of a lab arrangement are:

- Students at the computer do not miss group instruction or disturb the class when they take turns.
- All students get a turn at the computer.
- The lab arrangement is the most efficient use of computer time (Becker, 1984).
- Practice comes immediately after teacher instruction.

In both arrangements the practice could be accomplished by a combination of tutorials and games.

Evaluation

A keyboarding program should have both formative and summative evaluation questions. Typical formative evaluation questions are: "Is the program being implemented as planned?" "Are students practicing according to the schedule?" "Is teacher training for the program adequate?" "Is the software working?" "What program problems are solved and which problems remain to be resolved?"

A typical summative evaluation question is: "What percentage of the students met the exit criterion of 10 gwpm?" Other summative questions regard the quality of the student technique and the accuracy of the typing.

Information gained from the formative evaluation helps improve the ongoing program; information gained from the summative evaluation helps determine whether the program has accomplished its objectives.

Recommendations

If frequent and repeated use of the keyboard is required to achieve program goals, students must be proficient at the keyboard. Keyboard technique should support, not hinder, achievement of problem solving or composing. But establishing keyboarding skills takes time. Is keyboarding instruction worth the time in the elementary school? The answer is found by balancing classroom time required to achieve keyboarding skills against the advantages gained by rapid keyboard entry of data now and later. If the program described here is followed, the time requirement is 20 days of instruction for 35 minutes a day—less than 12 hours. The advantages are:

- More efficient student use of computer time;
- Focused attention on the task to be accomplished, rather than on keyboarding mechanics;
- Early learning and accomplishment of touch-typing technique.

Keyboarding instruction is an investment in more efficient learning. I think the scale tips in favor of teaching keyboarding before grade six. Keyboarding is becoming an increasingly important issue. Educators in every school district should resolve it in advance. **END■**

[Keith Wetzel, Center for Advanced Technology in Education, University of Oregon, Eugene, OR 97403.]

© Copyright 1985 ICCE

Bibliography

Beal, Jack, Lynn Churchill & Conn McQuinn. "State of Washington Computer Use Survey." University of Washington, 1983 (mimeographed).

Becker, Henry Jay. "The Social Context of Microcomputers: It's Not Just a Matter of Good Software." Presentation at the *AERA*, New Orleans. April, 1984.

Beverstock, Caroline. "Computer Using Educators, Arizona State University Conference, 1984," *6* (8), 9.

Brady, Holly. "Who Needs Ten-Finger Typing?" *Classroom Computer News*, Sept. 1984, 64-5.

"CCN Forum: Word Processing." *Classroom Computer News*, 1983, *3* (3), 24-7 & 74-76.

Cowles, Milly, Martha Hedley & Mabel Robinson. *An Analysis of Young Children Learning Keyboarding Skills*. Bethesda, MD: ERIC Document Reproduction Service, ED 238542, 1983.

Craighead, Donna & Mary Ellen Switzer. "Is Typing the Key to Computer Literacy?" *Instructor*, 1983, 93(2), 178-180.

Daggett, Gail, keyboarding instructor, Eugene, Oregon. Interview July 3, 1984.

Daiute, Collette et. al. "The Computer in the Writing Class: Problems and Potentials." Teachers College Columbia University: 1981 (mimeographed).

Gentner, Donald & Donald Norman. "The Typist's Touch." *Psychology Today*, 1984, *18* (3), 67-72.

Graham, Steve & Lamoine Miller. "Handwriting Research and Practice: A Unified Approach." *Focus on Exceptional Children*, *13* (2), 1980, 1-16.

Hendersen-Lancett, Lexie. "Typing: To Teach or Not to Teach." *COM 3*, 1984, *10* (3), 23-24.

Holmquist, Donna. "Business Education: Some Controversial Questions." *Business Education Forum*, 1983, *37* (5), 3-4.

Kercher, Lydia. "Life-Skill Learning for Preadolescents." *Business Education Forum*, 1983, *37* (9), 4.

Kennedy, Peggy. Formerly of Minnesota Curriculum Services Center, Whitebear Lake, MN. Telephone conversation, Nov. 9, 1984.

Kisner, Evelyn. "Keyboarding—A Must in Tomorrow's World." *The Computing Teacher*, 1984, *11* (6), 21-22.

Lindsay, Robert McDonald. *A Comparative Study of Teaching Typing Skills on Microcomputers*. Bethesda, MD: ERIC Document Reproduction Service, ED 220 597, 1983.

McClurg, Pat. "Keyboarding—An Issue in the Elementary School." University of Oregon, 1984.

Moursund, David. "The Two-Percent Solution." *The Computing Teacher*, 1984, *11* (7), 3-5.

Ownby, Arnola & Heidi Perreault. "Keyboarding in Kindergarten—Is It Elementary?" *Business Education Forum*, 1982, *37* (3), 19-20.

Russon, Allien & S.J. Wanous. *Philosophy and Psychology of Teaching Typewriting*. Chicago: South-Western Publishing Co., 1960.

Schmidt, B. June & Jeffrey T. Stewart. "Microcomputer Typewriting in Business Education." *Business Education Forum*, 1983, *37* (6), 23-32.

Schmidt, B. June. *Keyboarding: The State of the Art*. Bethesda, MD: ERIC Document Reproduction Service, ED 236 352, 1983.

Stewart, Jane & Buford Jones. "Keyboarding Instruction: Elementary School Options." *Business Education Forum*, 1983, *37* (7), 11-12.

Stoker, John. University of Oregon, Eugene, Oregon. Interview, Nov. 9, 1984.

Sunkel, Mary Jane & Martha Cooper. "Typewriting: Teaching Touch Keyboarding on Computer Terminals." *Business Education Forum*, 1982, *37* (1), 18-21.

Wetzel, Keith. "Formative Evaluation of the QUILL Program." University of Oregon, 1984 (typewritten).

© *Copyright 1985 ICCE*

Keyboarding—A Must in Tomorrow's World

by
Evelyn Kisner

Article after article has been written about using computers for educational instruction. These articles provide helpful suggestions and directions for using computers as instructional tools. Yet one very important factor in learning to use a computer efficiently has not been emphasized enough—*keyboarding!*

Just What Is Keyboarding?

Keyboarding is the term used to describe the activity of entering information on electronic equipment through the use of a typewriter-like keyboard. Keyboarding is the skill that enables an individual to use a computer terminal or microcomputer efficiently.

Who Needs Keyboarding?

Almost everyone! Projected use of keyboard terminals at all levels of activity—at home, school, work and recreation—clearly indicates that keyboarding is becoming a necessity.

The increasing impact of computers as an instructional tool emphasizes the need for all students to learn to operate an electronic keyboard, by touch, if they are to function effectively. Without keyboarding skills, students waste valuable learning time searching for the right keys.

Some institutions of higher learning are refusing computer time to students until they have acquired keyboarding skills. The reason: Computer time is too costly to be wasted while some students hunt-and-peck.

Is Organized Instruction in Keyboarding Needed?

Yes! Keyboarding is a psychomotor skill, and correct techniques are necessary for mastering such a skill. As students begin using the keyboard, they form lifelong habits where correct techniques are extremely important for efficiency. The most effective way to develop keyboarding skill is through organized instruction.

Although the two-finger, hunt-and-peck method may work for short response, such a method is not adequate when more lengthy keying is necessary. The two-finger habit requires looking back and forth from copy or instructions to the keys, searching for the right keys, then looking at the screen and back to copy. This habit is time consuming and, once formed, is difficult if not impossible to correct.

When Should Keyboarding Be Introduced?

Keyboarding should be taught prior to the grade level where students begin using computers for academic work, with a goal of mastery of the keyboard before students begin working with instructional software programs. Being able to keyboard, by touch, enables students to concentrate on instructional responses. (Suggestions for competence levels for different grade levels are provided below.)

Who Should Teach Keyboarding?

Although elementary teachers may be able to keyboard through transfer of typewriting skill, it is doubtful that any are prepared to teach keyboarding.

"Keyboarding should be taught prior to the grade level where students begin using computers for academic work, with a goal of mastery of the keyboard before students begin working with software programs of instruction."

Business teachers are currently the only group with the background preparation for teaching keyboarding. They have studied the psychology and methods for teaching psychomotor skills and have been applying those methods to teaching the typewriter keyboard.

How Should Keyboarding Instruction Be Implemented?

Some school districts preparing for extensive use of computers for instruction at the elementary level have developed workable solutions for meeting the keyboarding need.

Several options have resulted. In Omaha, Nebraska,[1] a conclusion was reached that a team-teaching approach with business and elementary teachers working together to develop instruction for keyboarding could result in the most significant gains.

The following options, with business teachers as instructors, were developed in Lincoln, Nebraska:[1]

- Elective courses in summer school with a tuition charge and parents providing transportation to a school where electric typewriters are available.
- Evening or late afternoon classes at a high school where electric typewriters are available. Parents would provide transportation.
- Taking machines and teacher to the elementary school. This would require a traveling teacher and a room of electric typewriters or microcomputers. The equipment and teacher would remain in a school four or five weeks, and then be moved to another school. With class periods of 30 minutes each day, students who wanted keyboarding skills could be accommodated.
- In some areas, smaller schools might share a traveling van equipped with furniture and electronic typewriters or microcomputers. This option utilizes a traveling teacher. One advantage of this option is that the furniture would be of suitable size and height for elementary students.

What Level of Keyboarding Efficiency Is Needed?

The objective of initial instruction in keyboarding for students at the elementary level should be mastery of the keyboard, by touch, with mastery to be achieved through:

- Correct posture at the machine.
- Proper techniques for striking keys.
- Accurate fingering of keys.
- Operation of basic machine parts.

For kindergarten through Grade 3, where students' input is generally numbers and one-word responses, achievement of 20 words a minute with accuracy could be considered adequate for responding to software instructions. Most students should be able to achieve this rate in four or five weeks.

For Grades 4-6, where input generally calls for complete sentences, the achievement goal might be set at 25 words a minute, with accuracy.

The financial and scholastic challenge of meeting elementary students' need for keyboarding skills requires support from the principal, the superintendent and the school board. The development and implementation of keyboarding instruction depends upon the cooperation of elementary teachers who have expertise in teaching elementary students and business teachers who have expertise in teaching keyboarding skills. Teamwork can result in an appropriate program for meeting the challenge.

END■

[Evelyn Kisner, Director, Business Education, Texas Education Agency, 201 E. Eleventh St., Austin, TX 78701.]

[1]Verda C. Rauch and Patricia B. Yanke, "Keyboarding in Kindergarten—Is It Elementary?" *Business Education Forum*, December 1982, pp. 19-22.

[2]Jane Stewart and Buford W. Jones, "Keyboarding Instructions: Elementary School Options," *Business Education Forum*, April 1983, pp. 11-12.

Robert Shostak • Lester S. Golub

One of the recurring dreams of every English teacher is to be able to sit down with a friendly machine capable of reading a stack of 125 compositions in a matter of minutes and spewing out grades unerringly. I know I had the same dream late in the '60s when working as a graduate student with Ellis B. Page on Project Essay Grade—a well-designed piece of research to determine if a computer could grade a writing selection as well as a team of English teachers. Needless to say the results we achieved did not realize my dream. The technology revolution has given rise to renewed interest in making this dream a reality.

However, in the article which follows, Robert Hertz seriously challenges the whole idea of using computers to teach composition. And he especially questions the use of the computer to correct and grade writing. It is likely that not everyone will agree with what Hertz says. We would welcome reader responses to his thesis for later publication in BITS 'N PIECES.

Robert Shostak, Florida International Univ., School of Education, Tamiami Campus, Miami, FL 33199
Lester S. Golub, Penn State Univ., College of Educ., 145 Chambers Bldg., University Park, PA 16802

Problems of Computer-Assisted Instruction in Composition

by
Robert M. Hertz

Why not program computers to criticize and correct student essays? It has been done, but not well and certainly not economically. While the idea may appeal to English teachers overburdened by endless stacks of compositions, there are several impediments.

A computer cannot criticize an essay on a particular subject unless it has been specifically programmed to deal with that topic. Programming involving meaning is very time consuming and expensive, and such programming requires an enormous amount of computer memory. A dictionary must be developed in order for the program to determine whether or not the diction is appropriate.

Scholars in the area of computational linguistics undertake large scale research projects to get a program that will be able to read and summarize a newspaper article dealing with a single topic.[1]

Even spelling cannot be separated from meaning. Computerized editing systems check spelling by comparing each word in the text against a list of, for example, the fifty thousand or one hundred thousand most common words. If the word does not appear on the list, the system tags it as a misspelled word. However, one major newspaper with a computerized spelling system printed "where father new best" for "where father knew best" because both "new" and "knew" appeared in the computer's list of spelling words.

Checking the grammar of a student composition or any text is an even more ambitious undertaking, one that is still in the experimental stages.

> IBM's approach is to do a grammatical parsing of each sentence. . . . When it cannot complete a grammatical analysis of a sentence, IBM's program, called EPISTLE, looks for one of 14 errors that it knows.
>
> On a test of 8,400 sentences in ordinary prose, the program correctly parsed about 60% of the sentences.
>
> Though its designers would eventually like to be able to ask the machine what a passage it has just read is about, they have had their hands full so far just trying to get the syntactic analysis to work.[2]

A student who gets 60% is passing, but just barely. Having a computer read and criticize student essays is complicated by the various factors that go into a composition. Criticism involves, among other things, judging whether the style, tone and level of usage are appropriate to the content, to the audience, and to the level at which the student is supposed to be writing.

There has been one particularly interesting attempt to get a computer to criticize and rewrite material. This is the Writer's Workbench, developed by the Bell Laboratories.

The Bell engineers call their computer the Writer's Workbench. They might have called it Strunk and White or Theodore Bernstein because both *The Elements of Style* and *The Careful Writer* by those authors-on-authoring have been programmed into the Workbench.

The Workbench, in effect, is a mechanical literary critic. A person feeds in his or her tortured prose and out pop suggestions such as you haven't heard since Miss Totman tut-tutted over your sixth-grade English papers. Simplify! Use "use" instead of "utilize." And stop splitting all those infinitives, or else.

Then the Workbench, like Miss Totman, gives one final sniff and grades you. The editors of *Discovery* magazine had the clever idea awhile back of submitting for judgment the Gettysburg Address and the first paragraph of *A Tale of Two Cities*. The Workbench was not pleased. It ordered Lincoln to shorten his sentences—an average of 26.7 words in length—to something closer to 15. "Your document," the Workbench advised him, "contains many more complex sentences than is common for this type of text."

But Lincoln got off lightly compared to Dickens, whose sentences proved to be 100% "complex." The Workbench had a couple of helpful suggestions. Instead of reading, "It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness. . . .", the opening sentence might run: "The times were the best and the worst, wise and foolish."[3]

"Having a computer read and criticize student essays is complicated by the various factors that go into a composition. Criticism involves, among other things, judging whether the style, tone and level of usage are appropriate to the content, to the audience, and to the level at which the student is supposed to be writing."

The developers of the Writer's Workbench have built into the software the ideals of writing contained in two standard texts, *The Elements of Style* and *The Careful Writer*. It is against these standards that the program makes its value judgments about the sentences and texts submitted to it. While it can look for things such as sentences of excessive length, no doubt by merely doing a word count for each sentence, it cannot spot grammatical errors, nor does it deal at all with meaning. It can determine if the writer is using too many simple sentences or, as in Dickens' writing, too many complex sentences. A computer can make value judgments only on the basis of the parameters built into the software as yardsticks against which the writing is to be measured. Any software unable to deal with meaning to some extent will accept a sentence such as

> John knows his mother-in-law better than the Pythagorean Theorem.

which is grammatically correct but logically nonsensical and semantically anomalous.

Even if these impediments can be overcome, we English teachers must ask whether we really want a program that evaluates writing for style, grammar or meaning. The goal, no doubt, would be a program that can consider all three aspects of writing, but as Patricia Gingrich, a researcher on the Writer's Workbench project, stated, "We feel that meaning is a very human feature. . . . We feel more comfortable right now with leaving the meaning and the organization up to the writer."[4] Would teachers feel comfortable leaving meaning and organization up to the student? They would have to, or mark those aspects of the paper by hand.

Consider the matter of style. If software scans sentences to see if they are passive and admonishes the student on the matter, what have we really gained? While overuse of the passive is considered a detriment to style, the passive is a legitimate structure of English and many other languages. It does serve useful functions, among them emphasizing the object noun phrase that has been moved into the subject position. A program that cannot reference the intent of a passage can only allow the writer a certain quota of passive sentences, and what this quota will be is subjectively determined by the preferences of the programmer. Suppose our student writer stays within his quota but uses the passive inappropriately in every passive sentence that he writes, and suppose that a second student uses the passive appropriately but exceeds her quota slightly. Which writer is going to be marked down? In this instance, it will be the second writer.

There is another style-related matter to consider—regimentation. Even if the parameters of style can be embodied in courseware, using it might constrain such software in the range of acceptable writing styles until literacy means the ability to write in the "official" style. Everybody's writing would look alike! Good teachers of composition do not impose their own styles on students, but help them develop writing styles that manifest their own personalities.

Style cannot be divorced from content. Yet, an essay-grading program that attempts to evaluate

style without considering meaning would ignore the appropriateness of style for the subject matter and would, in essence, be advocating one style for all situations. But, let us say that a program could be developed that would be able to correlate the style and the subject matter. Would it be able to determine if the writer is experimenting by using an elevated style with a mundane subject, as Pope did in *Rape of the Lock*, to satirize the absurdity of the subject? No, poor Alexander would be marked down for inappropriate style.

In short, can the entire gamut of literary criticism be programmed into software? Even if that is possible, which is doubtful, there is still the question of whether we should, for then we have set limits beyond which the art of writing cannot venture. Expository writing is creative, or at least should be, but the creative aspects manifest themselves in a different and perhaps less obvious way than in "creative" writing. Perhaps student compositions are as deadly as they too often are because the instructor does not expect the student to exercise imagination or creativity, and the student obliges.

"Even if the parameters of style can be embodied in courseware, using it might constrain such software in the range of acceptable writing styles until literacy means the ability to write in the "official" style. Everybody's writing would look alike!"

My position is that even if software could be developed to mark compositions, writing should be taught by a teacher and not by CAI. A subject that is open-ended and not clearly delineated does not lend itself to CAI. Argumentation and creativity are two very important aspects of writing that are certainly not limited domains; hence, they are inappropriate for computerization. Since style is intimately linked to content, tone and attitude, we cannot have software evaluate style, at least not without divorcing it from consideration of these other facets of writing. Since style, in turn, is composed of diction and the writer's selection of certain favored grammatical structures from all the possibilities that the language offers, we cannot have software evaluate grammar for appropriateness to style, although we can have it search for gross errors such as subject-verb agreement or use of "it's" for "its."

Does this mean that CAI must abandon the field of composition altogether? No, but its role seems limited to those facets of composition that are clearly delineated, such as sentence structure or the mechanics of grammar usage—not content or style. In this limited capacity, CAI can be of great service to both the instructor and students. Virtually every composition teacher has had the experience of explaining the mechanics of writing, receiving the compositions and discovering that the essays all had mechanical errors, but different errors. Since all students did not share the same problems, repeating a lesson on subject-verb agreement would benefit some students but would bore others. What is needed is a separate tutor for each student so that each student's time and attention span can be used only for those areas in which s/he needs additional work. Since CAI allows just this possibility, the student who needs more work on subject-verb agreement can learn that material with suitable courseware, while the student who has subject-verb agreement mastered can work on the dangling participles that are plaguing his or her writing.

By having CAI take over the teaching of the mechanics of writing, teachers can devote their energies to the critical thinking that distinguishes good compositions from those that merely lack mechanical errors but otherwise really say nothing. Teachers are needed to implant critical thinking and argumentation; attempting to program CAI courseware for this function seems more of a *tour de force* than a legitimate application of CAI. The present use of teachers' time to teach mechanics at the expense of teaching critical thinking has yielded the wooden compositions that prove that it is impossible to take nothing and say it well. END■

Robert M. Hertz, Associate Professor, English Dept., Humanities, California State University—Long Beach, Long Beach, California.

References

1. Roger C. Schank, Michael Lebowitz and Lawrence Birnbaum, "An Integrated Understander," *American Journal of Computational Linguistics*, (January-March, 1980) VI, 13-30.
2. Lee Dembart, "Computers as Writers—Some Bad Reviews," *Los Angeles Times*, 1 January 1982, p. 1.
3. Melvin Maddocks, "Now It's Reading, Writing . . . and Workbench," *Los Angeles Times*, 17 December 1981, p. 28.
4. Dembart, p. 14.

COMPUTERS IN THE TEACHING OF ENGLISH

Robert Shostak • Lester S. Golub

In the March 1983 English Journal, *Rexford Brown draws several conclusions about the results of the most recent national assessment of literature* (Reading, Thinking and Writing, *Denver National Assessment of Educational Progress, 1981), which are worth careful consideration. Brown notes that "trends in reading, writing and literature cohere." The statistics used to mark these trends indicate that the percentage of students who have mastered higher order thinking skills such as analysis, inference, interpretation and evaluation are declining. If we accept the national assessment as a valid indication of our students' inability to think critically, then Brown suggests that to correct the situation we ought to be certain we are helping students master the skills of higher order thinking in our classrooms.*

In the article which follows, Elaine Jarchow describes how SOFT-LIT, a microcomputer program she designed, can help students begin to acquire specific higher order thinking skills as they apply to the study of literature.

[Dr. Robert Shostak, Florida International University, School of Education, Tamiami Campus, Miami, FL 33199. Dr. Lester Golub, Dean, School of Education, Baruch College, City University of New York, 17 Lexington Ave., New York, NY 10010.]

Teaching Literature with the Help of Microcomputers

by
Elaine McNally Jarchow

During the late 60s and early 70s, I was a secondary school English teacher. In addition to teaching the classics associated with certain grade levels, e.g., *Julius Caesar* in the sophomore year, *The Red Badge of Courage* in the junior year, I also tried to introduce students to the modern classics. We read and discussed *To Kill a Mockingbird, A Separate Peace* and *Lord of the Flies*. I introduced a course in The Modern Novel where we read and discussed *They Shoot Horses, Don't They?, Catch-22* and *Catcher in the Rye*. (Today I would be certain to include the works of important writers such as Richard Wright, Maya D'Angelou, Lawrence Yep and N. Scott Momaday.) Many times during these experiences I asked myself if my students were getting as much out of these discussions as I was. The truth is that I never really knew whether all students were thinking as deeply about meaning as those who managed to monopolize class discussions.

When the educational computing revolution began, I examined much of the software in the language arts area and concluded that there was very little available for teaching literature. Although *Compupoem* and *Haiku* offered students an opportunity to try their hand at various types of poetry, and *Storybuilder* and *Storystarter* gave students a chance to practice their sentence combining skills, I could find no software which might be used to teach specific works of literature.

So two years ago I began to develop a software program which would engage students in exciting discussion of literary works. I pictured the high school senior who had completed *Hamlet* sitting at the monitor. "Did Hamlet love Ophelia?" prompts the computer. "No, he sent her away to a nunnery," responds the student. "Why do you think Hamlet was so upset at Ophelia's funeral, then?" inquires the computer.

> ***"I hoped that a series of higher-order, open-ended questions would greatly enhance each student's opportunity to draw inferences from facts. I believed that the floppy disk would help engage every student in a level of inquiry which might be lacking in a general classroom discussion."***

I hoped that a series of higher-order, open-ended questions would greatly enhance *each* student's opportunity to draw inferences from facts. I believed that the floppy disk would help engage every student in a level of inquiry which might be lacking in a general classroom discussion.

My plan for using the microcomputer to teach literature resulted in *Soft-Lit*, the surrogate literature teacher. An inquiry approach to great books, *Soft-Lit* for *Huckleberry Finn* begins with the following frame:

> Congratulations! You have just completed one of the great books of all time—THE ADVENTURES OF HUCKLEBERRY FINN. Now that you've finished, here's an unusual way for you to bring together the ideas and events that you have read about.
> Press SPACE BAR to continue

With *Soft-Lit* students can choose from three kinds of questions:

- Questions that test their ability to recall facts;
- Questions that test their ability to interpret facts; and
- Questions that help them draw inferences from facts.

Four elements of the novel—character, theme, setting and plot—provide the focus for each of the three kinds of questions. Students receive feedback on their answers, a summary of their progress and an opportunity to correct wrong answers.

The following are examples of the 40 questions from the first level of questions:

1. In the final adventure, Huck shows his loyalty to Jim by deeds rather than words.
   A. Yes
   B. No
   If A. Correct!
   If B. Incorrect. Huck and Jim together design the plot that attempts to free Jim.
2. *Huckleberry Finn* contains many comments that are critical of society in general.
   A. Yes
   B. No
   If A. Correct!
   If B. Incorrect. The novel reveals Twain's criticism of society, including such practices as slavery.

The second level of questions is designed to test the students' ability to interpret facts. The following questions are illustrative of the 40 questions in this level:

1. Jim runs away from his owner because
   A. Jim has been whipped.
   B. Jim's wife lives in the South.
   C. His owner is going to sell him.
   If C. Correct!
   If A, B. Incorrect. Jim runs away because his owner is going to sell him.
2. The smallpox story shows
   A. Huck's knowledge of human nature.
   B. How people help one another.
   C. What Huck thinks of doctors.
   If A. Correct!
   If B, C. Incorrect. The smallpox story is an example of Huck's knowledge of human nature.

The third level of questions offers a student the chance to draw inferences from facts. For example, the computer asks, "What does Huck mean when he says, 'Human beings can be awfully cruel to one another?'" The student chooses a focus for inquiry and builds upon evidence from the novel. The student's responses are then summarized by the computer, the results appearing on the screen in a succinct paragraph. The following example illustrates the branching effect of this level.

The student's replies are circled.

> What is Huck's major strength?
> A. Huck is always open to new experiences.
> (B.) Huck believes that conscience rather than social conformity should be the first rule of human behavior.
> C. Huck is a good student of human nature.

> The episode that does the most to stir Huck's conscience is
> A. His experience during his travels with Jim.
> B. His experience with the Grangerfords.
> (C.) His experience with the duke and king.

> What does Huck's experience with the duke and the king reveal to his conscience?
> A. That lying and cheating are unacceptable behaviors.
> (B.) That it is wrong to masquerade behind false titles.
> C. That people who are easily tricked need to be looked after.

"What is Huck's major strength?" That was the original question on which the other questions and statements were based. Now let's look at a brief paragraph summarizing your responses.

Huck's major strength is that he believes conscience rather than social conformity should be the first rule of human behavior. His conscience is stirred by episodes with the duke and the king. Huck's experience with the duke and the king reveals to his conscience that it is wrong to masquerade behind false titles.

How can secondary language arts teachers most effectively use *Soft-Lit?* I would suggest that the program can be used in six ways:

1. Learning centers can be created in the classroom. Students at three of the centers can deal with each of the three *Soft-Lit* levels of questions.
2. The teacher can use *Soft-Lit* during a lecture to enhance student understanding of character, theme, setting and plot.
3. Classroom discussions can be enlivened through debates on the third level of questions. For example, the question, "What does the river in *Huckleberry Finn* symbolize," branches into twenty-seven possible outcomes.
4. Teachers can develop extra credit projects which involve the use of *Soft-Lit*. For example,

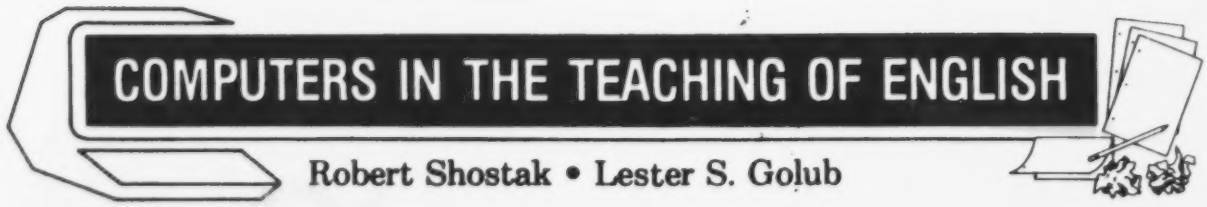
   > After you and your classmates have used *Huck Finn Soft-Lit*, prepare a panel discussion on the topic, "The Study of *Huckleberry Finn* is improved with the use of the microcomputer."

5. Before the Unit Test is administered, the teacher can assign students the task of reviewing by working through the three levels of *Soft-Lit* questions. This approach is particularly effective when the microcomputers and software are kept in the library or instructional resource center.
6. Healthy in-class competition can be fostered by dividing the class into teams of four members each. Each team member is then responsible for one of the focus areas—character, theme, setting and plot. Keeping track of team scores and working through the various levels of questions can help reinforce the key concepts in the novel.

That so many of our students have turned entirely away from reading classics to inserting flóppy disks is unfortunate. I hope using the floppy disk as the powerful motivator that it is will reawaken in students both a desire to read and to comprehend what they have read. I believe that the use of *Soft-Lit* will help students to write more effectively about literature. *Each* student will have an opportunity to probe for meaning in the completed novel.

[Ed. note: Soft-Lit *for* The Adventures of Huckleberry Finn *is available from the Perfection Form Company, Logan, IA 51546. At present it works on Apple II, II+ and IIe and includes disk and back-up, a teacher's guide and 3-ring notebook for $49.95. Several more titles will be available by January 1984.]* END■

[Dr. Elaine McNally Jarchow, Associate Professor of Secondary Education, N165C Quad, Iowa State University, Ames, IA 50011.]



[Editor's Note: While English teachers fortunate enough to have access to microcomputers to use with their students agonize over the dearth of good software, some of their colleagues are creating effective "programs" of their own with help from an unexpected source—the simple word processor.

These teachers, who have already discovered the benefits of using the word processor for composing and editing, are now finding exciting new applications for what many of us believed was a single-purpose piece of software. In the article which follows, Karen Piper describes how she has capitalized on some of the word processor's basic capabilities and developed "programs" to generate structured writing exercises and encourage creative writing. I have a feeling that after reading about Karen's efforts, many of you will not only want to try writing similar exercises but will also discover other useful applications. Why not share them with other TCT readers?]

The Electronic Writing Machine: Using Word Processors with Students

by
Karen L. Piper

Tempting students to become writers is not easy because learning to write well requires practice and hard work. You can make the writing act easier for your students, however, by the instructional use of word processors. Writing instruction using word processors will still require practice, but the mechanical process of writing can be eased. Students of all ages can benefit from the use of a tool that makes writing less cumbersome and more motivating.

The word processor allows students to write and revise with ease. Several aspects of writing with word processors contribute to this effect:

- Writing with word processors allows students to make insertions and deletions quickly and easily.
- Students can move words, phrases, sentences or paragraphs without messy erasures and scribbling.
- Drafts can be printed out, allowing students to edit clean copies and revise on the screen at a later time.
- Students don't experience "writer's cramp."

Using the Electronic Writing Tool

Many instructional uses exist for word processing programs. Students can use the word processor to complete creative writing assignments. Students do such writing more fluently as they develop writing techniques. I recently explored the use of word processing programs to deliver sentence combining instruction and encourage creative writing through story expansion. This technique allows students to practice manipulating written language structures in an effort to expand certain areas of writing ability.

With the guidance of Dr. Michael Angelotti, Texas Tech University, I completed an eight-week research study with fifth graders in Abernathy, Texas. I used a word processor to develop sentence-combining exercises for student completion and expansion. I wrote "clusters" of kernel sentences, which the students manipulated into one "writeout" per cluster—a sentence that combines all of the thoughts present in the cluster. To do so, the students delete redundant information; add conjunctions and commas; combine independent and dependent clauses; embed prepositional phrases, adjectives and adverbs; and make decisions concerning word order and choice.

For example: students are to combine the following cluster:

1. The coach was strong.
2. The coach was smart.
3. The coach encouraged her players.

To do so, students might embed and combine to create the sentence "writeout":

The strong, smart coach encouraged her players.

Once the combination is complete, students check the sentence to make sure the original meanings are present. In many cases, there is more than one correct writeout, and students are encouraged to try various approaches and manipulate the language to find the "writeout" that reads best to them.

Each exercise consists of approximately five clusters. A specific theme or story line runs through each exercise. As students complete the exercise, they expand the resulting writeouts into a story of their own.

The results of using microcomputer-delivered instruction in sentence combining and story expansion with fifth graders indicate that:

1. Fifth graders enjoy completing microcomputer-delivered sentence combining exercises.
2. Students eagerly expand the writeouts into their own stories.
3. Students who worked with sentence-combining exercises developed positive attitudes toward writing with the computer and toward sentence-combining activities.
4. Students who participated in the study improved in their writing ability, as measured by factors of writing maturity.
5. Students in the experimental study were highly motivated to write using the word processor.

Student Comments

Student comments concerning the sentence-combining activities of the microcomputer reveal specific points that are useful to teachers who wish to use the microcomputer as a writing tool. Some of these comments include:

"I liked that my hand did not get tired."

"It was easy to correct my mistakes."

"I liked writing on the computer because it was easier to erase and move things."

"It [the words on the screen] looks neater than my handwriting and I can find my mistakes better."

From the student comments and the overall results of the study, we conclude that the microcomputer can be used as an effective tool for structured writing instruction. For those of you who wish to try it in your own classroom, I offer a few tips:

1. Choose a word processing program that students can quickly and easily use.
2. Provide instruction in the use of the word processing program chosen.
3. Provide typing instruction or keyboard familiarity prior to using the word processor as a writing tool.
4. Train student assistants to help students get started with the word processor.
5. Prepare your text files carefully and well in advance. Know the objectives of your exercises and structure them accordingly.
6. Make sure you have a printer that will work with your microcomputer and word processing program.
7. And finally, be prepared to make arrangements for students who won't want to stop, or who will ask to come in early or stay late.

Given appropriate instruction and practice through activities such as sentence combining, students can become better, more motivated writers.

END■

[Karen L. Piper, Texas Tech University, P.O. Box 4560, Lubbock, TX 79409.]

[Dr. Robert Shostak, Florida International University, School of Education, Tamiami Campus, Miami, FL 33199. Dr. Lester Golub, Dean, School of Education, Baruch College, City University of New York, 17 Lexington Ave., New York, NY 10010.]

Robert Shostak

At the last NECC meeting, held in Dayton, Ohio, I attended a session offered by Bates Hoffer and Pat Semmes, both faculty members at Trinity University in San Antonio, Texas. I was attracted to the session because the presenters were going to talk about an experiment to produce haiku using Logo; and several years ago, I had developed a simple-minded program of my own which would generate haiku from input statements supplied by my students. What Hoffer and Semmes reported was so exciting that I asked them if they would share it with readers of this column.

What they describe in the article which follows demonstrates how powerful a tool Logo can be for teaching both basic skills and critical thinking. Moreover, the results of their "experiment" clearly illustrate that Logo is, indeed, a very powerful tool for the language arts teacher.

Computer Haiku:
A Classroom Experiment with Logo in the Language Arts

by
Bates Hoffer and Pat Semmes

INTRODUCTION

My freshman writing class used Logo to produce one of the haiku below during a two-week project. The other was found in an English haiku book. Which is which?

PELICANS — ALL DAY
DIVING INTO SCHOOLS OF FRY
TEEMING IN THE BAY

A SILVER MORNING
DRIFTING IN THE STILL SHADOW
SOFT SWIRLING SNOWFLAKES

DESCRIPTION OF THE PROJECT

The haiku from the project was submitted to the school literary magazine and accepted. The project also has been tried successfully with a 12- and a 14-year-old. The ease of the project suggests that fifth and sixth graders would also do well.

The goal of the project was the construction of a learning sequence in the language arts which paralleled the structure of the famous NIM project (see Harold Abelson's synopsis in *Apple Logo*, McGraw-Hill, 1983, pp. 103-113). There the students successively modify the tactics embedded in a Logo program, always coming closer to a complete set of tactics so that the computer can play Nim perfectly. In learning to come closer and closer to high-quality haiku, the students first learn to handle the problem-solving aspects of producing haiku with Logo and then go on to include language/literary features such as alliteration, assonance, word association, and so on. In this way they develop both major types of cognitive abilities—the problem solving, or quantitative, and the verbal, or qualitative.

The key to making the project work is Logo, a language so friendly that students require only a few minutes of instruction about how rules may be changed and a short period of experimentation before they can complete the project. One of the keys to the success of the project is that it is a *discovery process* for students. They are learning at every stage and are generating their own haiku, however primitive. In addition, they overcome any anxiety about making errors on a computer as they concentrate on eliminating bugs while working toward a solution. It is a good idea to copy and distribute the best haiku produced at each session.

Preliminary Session

The project itself began with a preliminary session giving seven major features of haiku: three lines of 5-7-5 syllables, a season word or suggestion of one, avoidance of personal pronouns, an emphasis on nature, an obvious or implied comparison or contrast of things or attributes, a Zen feeling communicated, and a sound/syntax feature which is tied to the meaning. Not all good Japanese haiku have all these features, but the best classic ones usually go beyond them. One classic haiku (translated from Japanese into English) is:

ON A WITHERED BRANCH
A SILENT CROW HAS SETTLED
AN AUTUMN NIGHTFALL

"Autumn, nightfall, silent, withered, settled" are mutually reinforcing images. The three nouns move from inanimate to animate to "heavenly." No human is mentioned, although the master poet Basho clearly conveys a human feeling/emotion that would be evoked by an experience depicted by the haiku. This popular haiku is often used to illustrate Japanese haiku.

Apple Lab Time

The first part of the project was a two-day session in the Apple Lab with three or four students per computer. The students received a handout on the haiku features and a few minutes of instruction on how to change rules written in Logo. Information about their project was already on the screen. On each disk was a set of four programs containing haiku-producing rules. Each successive program incorporated one new feature. The students were told how to add vocabulary to the rules of level one. In the class, students predictably began adding such non-haiku terms as "thermonuclear war." The next step was calling up haiku level 2 and figuring out the rule changes. The first step, based on Daniel Watt's *Learning with Logo* (McGraw-Hill, 1983, pp. 257-267), produced sets of three-word lines such as:

THE BOY RUNS
THE COMPUTER GOES
THE RABBIT HOPS.

The level 2 program changed the grammar of one line from article-noun-verb to article-adjective-noun. Within a few minutes, each team had found the change in grammar. The level 3 and 4 programs had similar minor changes. The students were expected to go beyond level 4 to produce higher-quality haiku in the second class period. The second part of the project was a 10-day period during which the students could go to the Apple lab at their convenience, refine the rules, and write a final report which included the best haiku produced

by their program. Questions at any stage were referred to either of the authors.

THE HAIKU RULES—Level 2

The team which produced the best haiku followed a sequence given below the rules listed here. These rules represent the relevant parts of level 2 of the project.

```
TO HAIKU2          (2 indicates 2nd
PRINT  SENTENCE1   level)
PRINT  SENTENCE2
PRINT  SENTENCE1   (which may duplicate #1)
END
```

```
TO SENTENCE1
OUTPUT ( SENTENCE ARTICLE NOUN
  VERB )

TO SENTENCE2
OUTPUT ( SENTENCE ARTICLE
  ADJECTIVE NOUN )

TO ARTICLE
OUTPUT PICKRANDOM :ARTICLELIST
END

TO NOUN
OUTPUT PICKRANDOM :NOUNLIST
END

TO VERB
OUTPUT PICKRANDOM :VERBLIST
END

TO ADJECTIVE
OUTPUT PICKRANDOM :ADJECTIVELIST
END

MAKE "ARTICLELIST [ A THE ONE ]
MAKE "NOUNLIST [ BOY COMPUTER
  RABBIT ]
MAKE "VERBLIST [ RUNS GOES
  HOPS ]
MAKE "ADJECTIVE [ RED GREEN
  TALL ]
```

The students first changed the nouns to those of one season: winter, snow, snowflakes, and so on. They also adjusted the adjectives to those which fit the season. When they found that the computer generated lines which repeated words, they rewrote the grammar of each line so that repeats would not occur. To keep track of the changes, they coded the types of words so that they reflected line and syllable count. NOUN1.2 represented a line-1 noun of two syllables. The NOUN1.2LIST received only appropriate nouns. Their final grammar structure was:

```
ARTICLE ADJECTIVE2 NOUN1.2
VERB PREPOSITION ARTICLE
  ADJECTIVE2.1 NOUN2.2
ADJECTIVE1 ADJECTIVE3.2 NOUN3.2
```

Just four random words for each line could produce over four million haiku! By this point the students had solved the problem-solving aspects of the project and could begin working on word associations. They eliminated "jack frost" and "sleigh bell," since those did not suggest "pure" nature. They picked verbs of movement for their winter theme to reflect the Zen influence of moving toward life. Finally, the winning team spent time working on assonance and consonance, especially alliteration. They wanted vowel sounds like those in "bit," "pot," "law" and "boat" to avoid the sound connotations of the high vowels. They wanted words starting with or containing "s" sounds to suggest the sound of winter wind. After inserting several acceptable words in each rule, they had the Apple produce several haiku and chose to submit the second one appearing at the start of this article. The "i" of "silver" is repeated twice in the second line, "o" is repeated in the last word of each line, and nine voiceless fricatives (six s's, one sh and two f's) all tie the haiku together with a euphony found in many classic Japanese haiku.

CONCLUSION

The students in the class, and those who have been involved in the project since, all expressed positive attitudes toward both the haiku problem and Logo. Those who had no previous experience with computers reported no computer anxiety by the second session. Three of their comments follow:

"I felt throughout that I was learning a great deal, not only about haiku but about the computer."

"I realized computers weren't so awful!"

"Now I want a computer to play with!" *[My favorite.]* **END■**

[Bates Hoffer, Chairman of the English Department, and Pat Semmes, Assistant Professor of Computing and Information Sciences at Trinity University, San Antonio, TX 78284. A disk which contains the project and support information may be obtained at cost from the authors; $6 to cover disk, handling and postage. Versions are available for Apple IIe or Commodore.]

Robert Shostak

Editor's Message

Teachers of composition know how important feedback is to the developing writer. We also know that the more specific our suggestions for corrections and revisions, the more likely the student writer is to improve. Unfortunately, large class sizes and the pressure of time often prevent us from being as thorough as we should be in order to produce the results we want.

Leigh Holmes, this month's contributor, has enlisted the computer in a most creative way to increase dramatically his effectiveness as a writing instructor. His *Follow 2* program allows him, with a few keystrokes, to make thorough comments on the most common mistakes made by student writers and to produce hard copy for them from a printer.

FOLLOW 2— A Theme Commentary Program

by

Leigh Howard Holmes

***Follow 2:* Computer-Assisted Theme Commentary**

To date, professional discussions about computers in English have focused on student-centered software such as word processors, prewriting aids, style analyzers and skill exercise programs. Yet many of us who work with large numbers of students have been motivated to develop software that assists instructors as well as students. A program of the latter type has enabled me to move frequently made essay comments to a printer by pushing numbers on a computer keyboard, thus reducing my grading time by at least 20 percent.

My own 32K theme commentary program, *Follow 2*, is presently on cassette for a 64K TRS-80 Color Computer, but it can be easily put on disk for TRS-80 models I-IV. While I use an Okidata Microline 92 printer, the program can drive other printers. I am working on an Apple IIe version.

Somewhat similar to theme commentary programs designed by Jack Jobst and William Marling, *Follow 2* requires that students number each line of their themes. The program is built around the idea that one can expect to see many features of students' writing repeated semester after semester, and that big advantages result from the application of programmed comments to these features, which are identified by typing in the line numbers from themes before calling up the comments. Programmed comments save my time and aid students, since they are specific and thorough and can therefore better help students understand their writing problems and how to go about solving them. "Live" comments addressing atypical, "nonprogrammable" theme problems can also be typed.

Two key features of such programs are how effectively the comments are organized and how well the comments apply to the things that frequently have to be said about student writing. I took these features into account. The main menu includes categories of items that English instructors are usually concerned with in student papers:

1. Specific Comments
2. Following Directions
3. Overall Structure and Development
4. General Usage
5. Mechanics Usage and Spelling
6. Quotations Usage and Research Mechanics
7. General Comments
8. Final Comments and Grade

Menu items 1 and 8 allow me to type live comments, and items 2 through 7 move me to submenus.

Each submenu offers a variety of choices. For example, typing a 4 calls up the General Usage submenu:

1. Point of View
2. Regional Usage
3. Subject-Verb Agreement
4. Verb Parts
5. Tense
6. Pronoun Usage
7. Next Menu
8. Comma Splice/Run-on
9. Fragments

Submenu 3, Overall Structure and Development, is:

1. Thesis
2. Theme Development
3. Paragraph Development
4. Recent Class Focus
5. Coherence and Transitions
6. Invention
7. Next Menu
8. Introduction and/or Conclusion

Each of these submenus (except Next Menu) has an additional layer of submenus, so *Follow 2* currently has 13 submenus that lead to comments. The number of possible comments is limited only by the computer system available. The main problem is how to manage all the potential messages: Organizing them in categories, coding them, and finding a way to arrange the codes to facilitate easy movement from student paper to entering the codes at the computer.

Follow 2's method of organizing these messages, a feature which distinguishes it from similar programs, is the use of an organizing mode—code sheets—which includes abbreviations of the messages beside one-, two- or three-digit numbers. The leftmost digit ties the message to the appropriate submenu item. The arrangement is:

S-V agr	3
Read HH; do ex	31
Read HH; do ex 2	32

Checking the 31 will eventually result in the student being asked to read some pages in the *Harbrace College Handbook* and do an exercise. Code 32 references a backup exercise.

In addition to saving time that would be spent calling up submenus, the code sheets have other advantages which are logistical rather than instructional. The sheets prevent terminal dependency, a problem noted by Bill Marling, author of *Grader* (mentioned in *Focus: Teaching the Language Arts*, Spring 1983, p. 51). If I want to grade papers away from my computer, I can check code sheet items to enter later at home. I can make nonprogrammed comments directly on the student's paper. Thus the student will have the usual printer-generated page and comments on the theme itself.

Since using the program, I have discovered six unanticipated instructional advantages—advantages which are improving my communications with students and hence, my teaching.

First, in addition to my written comments being more thorough, they are also unaffected by late-hour grading, which tended to result in sharper and more abbreviated comments.

Next, I provide the nuances of commentary which I had tended to neglect under time constraints. I now have 140 statements on call, with many other lines that can be added for a particular assign-

ment. Where I once communicated with "problems with thesis," I can now stipulate such things as "lack of a challenging assertion made by the thesis" or "absence of word repetition that would reinforce the thesis."

A third advantage is that, unlike the old-fashioned theme cover sheet checklist (which simply registers errors for the student to look at and perhaps forget immediately), the neatly formatted page encourages the student to read and follow up. The student is asked to note the assignment in a specially formatted space near my final comments. The assignment of exercises, an activity that sometimes seemed too ambitious to undertake late at night, may now be done quickly.

Fourth, being able to produce a lengthy statement by entering a number enables me to articulate course objectives to students frequently so that they can more easily understand the purposes of each assignment.

Another instructional advantage is automatic menu marking. There is often information that all students must receive, e.g., the acceptable format for writing papers. The code sheet marking allows me to respond to individual papers and to provide general announcements.

Finally, I have branched out in my communications with students. I plan printed statements for each new assignment in order to reinforce a recent course focus. Such statements will become a part of my course planning. When printed out they recall previous emphases, forecast upcoming course tasks, and cite the progress made by a class. They help provide a positive emphasis throughout any term.

I am convinced that we are going to read more discussions about commentary programs like *Follow 2*. Teacher-centered programs may never be as popular as student-centered ones because the student is really the focal point of the learning process and represents a much larger market for software publishers. However, theme commentary programs enhance our teaching and do so in such dramatic ways that they are bound to become a major professional interest of the eighties. They are fun, too. I fully exect many more unanticipated and pleasant teaching experiences in the coming months.

The program is still evolving. If *Follow 2* interests you, please write. I am willing to see the program disseminated, and I would like to investigate collective efforts to improve theme commentary programs. **END■**

[Dr. Leigh Howard Holmes, English Department, Cameron University, 2800 West Gore, Lawton, OK 73505.]

[Dr. Robert Shostak, Florida International University, School of Education, Tamiami Campus, Miami, FL 33199.]

Bank Street Writer— Does the Tool Define the Process or Vice Versa?

by
RANDALL BOONE

Word processing has become a new educational buzz word in the last couple of years, especially with educators who are wrestling with the issue of computer literacy in their schools. Student use of word processing software to compose, revise and print their writing is becoming standard in school curricula which offer computers. For teachers faced with the task of implementing computers into their classroom, it appears a safe, sure, first step into computer literacy for their students. This is perhaps true. Yet it seems that in this headlong rush to integrate computers into the curriculum, too little thought is being given to possible constraints which might lie hidden underneath the obvious advantages the computer provides.

Ask teachers about word processing and one program in particular, the *Bank Street Writer*, is likely to be a main topic of discussion. This word processing program, with its easy-to-learn, menu-driven format and secure handling of files, has virtually taken the educational market by storm. Teachers feel comfortable with this product; the name *Bank Street* itself implies educational excellence. The program is marketed to the educational community as being a facilitator of the composition process, and the idea seems sound enough. First, students write their thoughts without the hindrance of correcting errors or focusing on mechanics. When they have finished, then it is time to review the piece to revise both meaning and mechanics. This is the obvious advantage of word processing: changes can be made easily without the drudgery of rewriting the entire piece. Looking a bit deeper into the writing process, however, it becomes evident that *Bank Street Writer* may possibly impose serious constraints to a child's natural writing pattern.

Most writers will agree and indeed research confirms that writing is not a singular act, but rather an ongoing process consisting of several progressive stages. Rarely will any two authors' writing processes follow identical paths, yet three basic stages seem to occur in almost everyone's writing. The stages are most often categorized as 1) Prewriting, 2) Composing and 3) Revising. Research into the cognitive processes involved in writing suggests, however, that the writing process is not simply a linear progression through these stages. Rather, a model is defined which allows for interruptions during prewriting, composing or revising. Each of these stages is a recursive implementation of the same three stages. Revision and prewriting, for instance, may well interrupt composing, the middle stage, quite often. The reason for this is simple: writing is generative. That is, as one writes, new ideas are generated by the act of writing. This may cause a writer to want to make changes halfway through a story, a paragraph, or even in mid-sentence. It is at this point in the composition process that *Bank Street Writer* fails to accommodate the writer.

It would appear that the developers of *Bank Street Writer* chose a model of writing which requires the writer to compose an entire piece in one "stream-of-thought" burst of writing, not allowing for the recursive/generative nature of the writing process. Simple revisions during the composing stage are difficult to accomplish using *Bank Street Writer* with its separate modes for cursor movement and text insertion. The important act of revision is effectively discouraged by the complexity of the routine involved in accomplishing it. Word processing software which allows the writer to both write and edit in one mode is much easier to use and therefore more likely to be used for revision.

The following is a step-by-step description of this slow, seven-step process for backtracking, making a change, and returning to writing using *Bank Street Writer*:

1. Press ESC for cursor mode.
2. Move cursor to desired position.
3. Press ESC for write mode.
4. Make change or insertion.
5. Press ESC for cursor mode.
6. Return to previous position.
7. Press ESC to return to write mode and continue writing.

The alternatives are not any more appealing. One may simply backspace thereby erasing everything up to the point where the change is to be made, or hope to remember what change to make when going through the piece during the revision stage.

What is happening here is subtle yet very important. The way students naturally think during a writing episode is possibly being reshaped by the constraints of the word processor. This may be especially prevalent with beginning writers in the early grades. If this is true, educators must look very carefully at the implications this might have for students as writers and as thinkers.

There is little doubt that the ease of revision which word processing allows can be of great benefit for students as they begin to explore language through their own writing. As educators, it is our responsibility to provide them with the most appropriate tool for this task. It is necessary to examine word processing software with an eye for how much freedom the writer will have in using it rather than how easily it may be learned.

[Randall Boone is a doctoral student in computer education at the University of Oregon. He has taught elementary school for nine years, including five years overseas in Indonesia and Saudi Arabia.]

"But What Do I Write?"— Literary Analysis Made Easier

by
Helen J. Schwartz

When I teach an introductory literature course at Oakland University and announce the usual written assignments, I sympathize with students when they ask, "But what do I write?" Even when I explain in class and pass out sample papers, students sometimes have trouble getting started. Although there's nothing like individual tutoring and feedback for helping such students, many students (especially at a large commuter school) find it difficult to schedule a conference time with me or with peers that could get them going on a paper. Other students find a class discussion or individual conference too intimidating.

For such students, I've developed a CAI program called SEEN—the Seeing-Eye Elephant Network. The program helps a student develop a character analysis for literary studies. The Seeing-Eye section of the program is a tutorial which helps a student "perceive" what s/he has "seen"; here students are required to write, in response to prompts from the computer, a "notice" which is then posted on the "Network." The Network section of the program allows students to see and write "comments" on the "notices" posted on the "Network" by other students. The Elephant is the section of the program which records and remembers what the student has written (or revised) in a "notice" or in a "comment." These "notices" and "comments" are stored as textfiles which can be loaded onto an Apple-compatible word processing system for printing or revision into an essay.

The program is based on the principle that learning should be *active*—the learning should take place in the *student*, not the programmer. Therefore, the program does not judge the responses of the student—either in the tutorial or on the network. The aims of the program are:

1. to provide students with thought-provoking prompts so that they can generate ideas,
2. to encourage the refining and expansion of their ideas by giving and getting feedback (anonymity is provided through the use of pen-names),
3. to encourage students to learn *how* they have learned by providing a record that traces the development of their ideas.

Let's see how one student actually used the program in her world literature class. During a time scheduled for her convenience, she responded to the program's prompts, hypothesizing that Satan in *Paradise Lost* is "tricky". The tutorial then asked Kami (her pen-name) to provide various kinds of evidence to support her hypothesis. (The computer's questions are in capital letters and Kami's answers follow.)

WHAT DOES SATAN DO THAT SHOWS SATAN IS TRICKY?

1. He enters the body of a serpent to disguise himself.
2. He convinces Eve to eat from the tree of wisdom by telling her lies about it.
3. He is able to convince Eve that he is a wise serpent and not her enemy.

The program then asked Kami for further evidence that Satan is tricky—because of 1) what he says, 2) how others react to him, 3) how his behavior compares to others in a similar situation, and 4) what the third-person narrator (if any) says to him.

Users get numerous chances to modify and edit a "notice." The program does not judge spelling or ideas, but it does, at this point, encourage self-criticism by asking students to look at evidence critically. For example:

OK, NOW TRY TO THINK OF ANY EXCEPTIONS OR CONTRADICTORY EXAMPLES.

WHAT EVIDENCE SHOWS THAT SATAN IS NOT TRICKY?

Adam and Eve knew that there is someone out to get them.

AT THIS POINT DO YOU WANT TO
1 = CHANGE LIST ENTRY (& ERASE EXCEPTION JUST LISTED)
2 = EXPLAIN THE APPARENT CONTRADICTION
3 = ADD THE EXCEPTION WITHOUT COMMENT

DO YOU WANT 1, 2, OR 3?
2 [carriage return]

COMPLETE THE FOLLOWING STATEMENT:

ADAM AND EVEN KNOW THAT THERE IS SOMEONE OUT TO GET THEM BUT . . .

They don't know what he will do or what he will look like.

When Kami completes this part of the tutorial, she has already generated enough ideas to begin an essay on Satan.

As the example demonstrates, this CAI program lends itself to various analytical subjects. What follows outlines the general format of the analytical approach, the application for character analysis and selections from Kami's actual responses:

General format	Particular application with sample response in parentheses
[Pick an X]	Name a fictional character X in a literary work (X = Satan in *Paradise Lost*)
[Create hypothesis: X=Y]	Describe the character X by completing the following: X is _____Y_____. (Satan is tricky.)
[Argue that X=Y]	Provide evidence to show that X is Y:
[Regarding type A evidence]	What does Satan *do* that shows Satan is tricky? (He enters the body of a serpent to disguise himself.)
[Regarding type B evidence]	What does Satan *say* that shows Satan is tricky? (He tells Eve lies about the tree of wisdom to get her to eat from it. . .)
[Regarding type C evidence]	How do *others* react to Satan that shows Satan is tricky? (God had originally thrown Satan out of heaven because he didn't trust him.)
[Regarding type D evidence]	How does Satan *compare* to others in a similar situation and how does this show Satan is tricky? (The other angels do not compare themselves to God and are not thrown out.)*

*Note that the program does not (and cannot) respond to inaccurate or unclear responses. That is a job for *people*, who respond on the Network.

[Regarding type E evidence]	If there is a third person narrator, what does he say that shows Satan is tricky? (He says that Satan chooses the subtlest of beasts for his embodiment.)
[Consider conflicting evidence]	What evidence shows that Satan is *not* tricky? (Adam and Eve know that there is someone out to get them.) In light of this evidence, you can now revise your notice, explain the apparent contradiction or leave the exception without comment. [The student "explained" her contradiction by adding the following: (but they don't know what he will do or what he will look like.)]

The following questions have been added to SEEN after analyzing results of the field testing in which Kami took part.

[Does X change?]	Does the character *change* as time passes in the literary work, and if so, how?
[Summarize.]	How would you summarize your view of the character?

SAMPLE

SYLLABUS COMPUTERS IN COMPOSITION 3 CREDIT HOURS

REQUIRED READING:

Graves, D.H. Writing: Teachers and Children at Work. Heinemann Educational Books, 1983.
Evaluator's Guide For Word Processing Software
Packet of articles available from Krazy Kats copy service on Alder between 12th and 13th.

GENERAL OBJECTIVES: To assist pre-service, graduate students and teachers in developing an understanding of: the writing process; the importance of revision in all writing at all levels; the role a word processing program can play in altering the revision stage of the writing process; the implications and strategies which the microcomputer makes possible for writing instruction.

ASSUMPTIONS: This course is based upon the following assumptions:
(1) Writing is not a singular act, but rather an on-going process consisting of several progressive stages: pre-writing, composing, and revising being the most common categories.
(2) Meaningful revision to a piece of writing at both the "meaning" and "mechanics" levels is necessary for good composition.
(3) The ease with which revision can be accomplished with a word processing computer will make a significant difference in the way a person writes. This difference will be of a positive nature.
(4) People learn writing skills by writing--i.e., within the context of their own writing, students will best learn skills of both content and mechanics.

SELECTED SPECIFIC OBJECTIVES:

*To be able to recognize and discuss the different stages of the writing process.
*To be able to understand the importance of the revision stage of the writing process, in particular the roles of rewriting, editing, and conferencing.
*To be able to integrate the use of a word processing computer into a writing program at any grade level.
*To be able to foster the writing process approach through the use of a word processing computer.
*To be able to evaluate and select appropriate word processing software for a particular audience.
*To be able to critically evaluate educational software for writing instruction, and use those which are available creatively.
*To be familiar with several different word processing

COURSE REQUIREMENTS:

1) Write a final draft "creative" composition (weight 15%, due Oct. 18).

2) Evaluate a word processing program (weight 20%, due Oct. 30).

3) Prepare and teach a pre-writing activity to the class (Weight 20%, due Nov. 6-20).

4) Write a final synthesis paper on the teaching of writing with emphasis on Computer Technology (weight 25%, due Dec. 6).

5) Attend Class regularly.

6) Participate in group discussions of assigned reading materials.

7) Practice conferencing techniques with a group or individual in the class.

8) Preview 5 new writing software programs (#5, 6, 7, 8 weight 20%).

COMPUTERS IN COMPOSITION
CI 410G

DATE				
Sept.	27	Writing	Graves, pp.3-63	
Oct.	2	Prewriting	Graves pp.65-93, SR pp.1-12	
	4	Conferencing	Graves pp.97-117, SR pp.13	
	9	Conferencing	Graves pp.119-148, SR pp.14-16	
	11	Revising	Graves pp.151-170	
	16	Revising	Graves pp.171-203	
	18	Writer's Development	Graves pp. 205-229	
	23	Evaluating Word Processors	SR pp.17-22	
	25	Writing Research	SR pp.23-42	
	30	Writing REsearch	SR pp.43-58	
Nov.	1	Comp. in Composition Research	SR pp.59-69	
	6	Comp. in Composition Research	SR pp.70-81	
	8	Computer Uses in Writing	SR pp.82-98	
	13	Computer Uses in Writing	SR pp.99-100	
	15	Word Processors of the Future	SR pp.101-103	
	20	Preview of Writing Software		
	22	Thanksgiving Holiday		
	27	Comp. in Comp.: Prerequisities	SR pp.104-105	Holly Brad
	29	Comp. in Comp.: Prerequisities	SR pp.106-112	
Dec.	4	Comp. in Comp.: Advan. & Disad.	SR pp.113-125	
	6	Final Synthesis paper due		

*SR (Selected Readings Packet)

CI 410G
Computers in Composition
Fall Term, 1984
Review Guidelines

Write a word processor review which uses "ease of composing and revising" as a major criterion for evaluation. Your review should answer the question: Does the ABC word processor allow students to write easily or does it hinder writing? A writing teacher, parent or student should find your review helpful when considering this product. Review information includes:

1) product information
   - title
   - publisher
   - grade level
   - cost
   - hardware requirements
2) description
   - intended users
   - description of operation
   - ease of use
   - manuals and tutorials
3) balance tests
   - strengths vs. weaknesses
   - value vs. price
4) personal recommendations
   - students
   - teachers
   - parents

The "Evaluator's Guide to Word Processors" is a resource which lists all of the major categories that you may need to consider when evaluating a word processor. You may wish to answer the "Evaluator's Guide..." questions prior to writing your review (a pre-writing exercise). Two additional resources are in the selected readings (Boone pp.17-18 and Knapp pp.19-22).

<u>The review should be written as if it was for publication, and it will be evaluated accordingly.</u>

You may use the word processor of your choice to compose and revise your unified essay. Feel free to bring your word processing system to the lab for conferencing and writing.

BEST COPY AVAILABLE

CI 410G
Computer in Composition
Fall Term, 1984

Final Draft Personal Writing

1) Submit a folder which includes a copy of each draft of your composition. (Remove the tractor-feed margins on both sides of each page.) At the end of each draft, list the changes you decided to make in the next draft.

2) The final copy should be special. Use bonded paper when you print the final copy. Make certain that the printer has a good ribbon in it. Proof read the final copy carefully. Your final conference session may be used to polish the piece.

3) The due date will be Friday, Oct. 19, 5:00 p.m. You may place it in my mail box in CATE room 118.

COMPUTERS IN COMPOSITION
CI 4106
Fall Term, 1984

Presenting a Pre-writing Activity

You are to explore a pre-writing activity and present your findings to the class. Choose one of the following: 1) a specific computer program designed to help students gather thoughts prior to composing, 2) a general computer program which could be molded into a pre-writing activity, for example, a data based information activity, 3) another type of activity (non-computer) which could be used as a pre-writing activity, for example, a science experience.

Your product should be 1) a one to two page discussion of your findings, and 2) a 10-20 minute demonstration.

The questions that should be answered in your paper and demonstration are:

- -How does the program work?
- -Is is appropritate for several grade levels?
- -Would you use this activity in your classroom?
- -How does this tool assist in writing?
- -Is it based on a theory of writing?
- -Is this a good use of the computer?

Computers in Composition
CI 410G
Fall Term 1984
Synthesis Paper

The final examination for Computers in Composition is a synthesis paper. The purpose of the synthesis paper is for you to combine various aspects of computers and writing into a coherent whole. Your paper should include:

-A summary of a philosophy of writing of Graves/Russell/Murray as discussed in class.

-Your personal philosophy of writing based on both your experiences as a writer and as a teacher of writing (if applicable). Use the research on writing to support your positions.

-Your view of the myths and realities regarding the use of word processors to improve writing in the classroom, supported by your observations of students using word processors, discussions with classroom teachers and the research on word processors and writing.

-Discuss what you have learned in this class and how you intend to apply it in one of the following roles:

computer literacy teacher
parent
district computer coordinator
teacher trainer
author of writing textbook
elementary and middle school teacher
math teacher
computer club sponsor

<u>Deadlines</u>

Draft: Complete a draft by Nov. 27.
Bring two copies to class.

Comments: Write feedback on the papers of two colleagues.
Sign the papers you have commented on.
Return papers to the authors on Nov. 29.

Optional: Set appointment to have copy editor revise mechanics.

Due date: Final copy will be due in class on Dec. 6.

Guidelines: The paper should be eight to fifteen pages in length and it should be double spaced.
The final paper must be copied or printed on high quality bonded paper.
The grade on late papers will be reduced!

SOFTWARE REVIEWS

Reviews

Linda Rathje, Editor

Kidwriter (Spinnaker), ***Story Maker*** (Scholastic) and ***Bank Street Story Book*** (Mindscape) are three programs that enable young writers to create illustrated stories. Students can "make" pictures or scenes and develop accompanying stories that can be read on screen or in print. The reviewers were able to work with children as they explored these programs, and have shared their observations.

Three programs for preschool and primary grades are also included this month. Reviewer Christine Chaille sees the two Joyce Hakansson programs ***Ranch*** (Spinnaker) and ***Kermit's Electronic Storymaker*** (Simon & Schuster) as "construction kit" or "electronic tinkertoy" activities which provide the young child with opportunities for self-directed experimentation. C & C Software has incorporated a management system in their ***Letters and First Words*** program, and reviewer Scott Turner, a first grade teacher, particularly appreciates the opportunity to make program modifications.

Last, but certainly not least, is Broderbund's popular program, ***The Print Shop***. Educators should definitely examine this versatile graphics program.

KIDWRITER

Spinnaker Software Corp., 215 First Street, Cambridge, MA 02142
Grade/Target Age Group: Ages 6 to 10
Subject: Creative writing
Hardware Required: Apple II, II+, IIe or Commodore 64, IBM PC, IBM PCjr
Color monitor recommended; joystick optional; one disk drive
Cost: $29.95 (Apple), $29.95 (IBM), $26.95 (Commodore)
Reviewed by: Chrissie Forbes, Santa Clara Elementary School, Eugene, OR, Grades 1-2

Kidwriter is a combination word processor and picture maker for elementary age children. It follows the teaching format used in many creative writing lessons: Draw a picture and then write about it. What child faced with the task of writing a story wouldn't be motivated by the fun of creating a picture on the computer screen? Any program using this format offers a potential writing tool of special consideration for the elementary teacher.

How It Works

Choosing from 99 objects, the child first creates a picture, selecting the size, position and color for each object. One object is manipulated at a time and cannot be repositioned once the child adds the next object to the picture. When the picture is completed, seven lines remain on the screen to compose a narrative or description of the picture.

When a page is completed, the child may choose to: 1) make a next page to the story; 2) start a new story; or 3) store it on disk for later reading. Once a story page has been stored, it cannot be changed.

The commands needed for editing the text are not prompted on the screen but are easy to find in the manual. On the Apple IIe and Commodore 64, the cursor is controlled through the arrow keys. On the Apple II or II+ the cursor is controlled through the Control E, D, X and S keys. Erasing is accomplished with the Escape key, one letter at a time. The child may insert letters or words, and the rest of the text will automatically move to the right. The Apple IIe and Commodore 64 versions are *very* easy to work. Young students can work independently, making it ideal for classroom use.

THE CHILDREN LIVE IN A HOUSE ON THE
HILLSIDE WHERE FLOWERS GROW A LOT
AND A TREE GROWS THERE SPECIAL FOR
HALLOWEEN. THERE IS A JET THAT FLIES
OVER THE CHILDREN EVERY MORNING AT
NINE.

Kidwriter.

The commands for moving the objects in the pictures use the same keys used to edit on the Apple II or II+, which poses a serious problem for Apple II and Apple II+ owners. While Control E moves an object up higher on the screen, plain E erases the entire picture. While Control D moves the object to the right, plain D indicates that the child is done making the picture and ready to write the story. Once this D command is entered, changes cannot be made in the picture. My six-year-old daughter lost almost half of the pictures she made through this type of beginner's error. Choosing different keys for these commands or having an "Are you sure?" message could easily avoid these problems. (Apple IIe and Commodore 64 owners won't have this problem, because the picture objects are moved with the arrow keys.)

Strengths

Kidwriter is easy to use. I tried it out on my young daughter and her playmates, and with very little explanation they were busy taking turns drawing and composing. The short (ten-page) manual was simple to follow. It was also easy to find all the commands that might be needed. With a couple of exceptions, the program displays all the commands that a child will need up on the screen at the appropriate time.

With the range of objects, colors, sizes, positions and backgrounds to choose from, the picture-creating aspect of the program itself makes this an attractive product. Of course, if making a picture is all you're looking for, there are other more sophisticated (and more expensive) programs that allow for more creativity.

The manual is easy to use and any computer novice can experience mastery in no time. There are plenty of useful prompts and the user is asked to confirm all decisions to erase text.

Weaknesses

My daughter was frustrated by the accidental erasing of her pictures. A more glaring problem, from my viewpoint as an elementary school teacher, is the lack of editing options. Some children will want to write more than seven lines of text. When the seven lines are filled, the extra text is not automatically transferred to the next page of the "story." A sound beeps with each extra word and these words are lost. The program makes no allowance for the child who would like to write more and draw less. Also, once a child has developed a train of thought, it seems a shame to have to stop in mid-sentence to draw a new picture before being able to complete the idea.

In school (and at home) we try desperately to teach children that writing is a process involving many steps. We write, rewrite, write and then rewrite again. We move sentences around. We

BEST COPY AVAILABLE

BEST COPY AVAILABLE

change words. We add explanations and cross out unnecessary verbiage. We remember important information or even change a story's entire focus. Because this program doesn't allow the child to retrieve a page or an old story and make changes, it could reinforce the mistaken notion children get that writing is a one-step process. With all the attention given nationally to our children's inability to write and the need to teach these essential editing skills, I'm disappointed that *Kidwriter* doesn't provide for rewriting.

A final weakness of *Kidwriter* is the lack of a printing option. If a parent or teacher is hoping to print copies of a child's work for display or examination, then this program falls short. As a teacher I would insist on a program with a printing option so that my students could take their stories home.

Recommendation

Kidwriter was the first program to combine picture making and story writing. Newer programs are more sophisticated, but *Kidwriter* will give your child plenty of opportunity to "draw" pictures and compose stories. Because of its price, ease of use and motivating qualities, I recommend that primary teachers examine this program when considering combined word processor and picture-making software for elementary age children.

STORY MAKER

Scholastic Inc., 2931 East McCarty St., P.O. Box 7502, Jefferson City, MO 65102

Grade/Target Age Group: Ages 8 years and up

Subject: Creative writing and illustrated text editing

Hardware Required: Apple II+, IIe or IIc (64K RAM minimum), single or dual disk drive, KoalaPad, joystick or AppleMouse required.
Optional printers: Epson (with a Grappler card), ImageWriter or Scribe printer

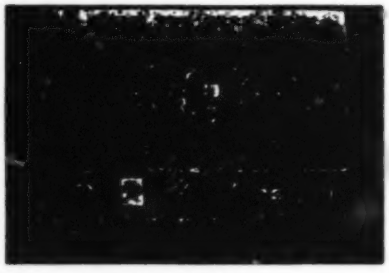
Cost: Personal edition—$39.95 (one program disk, one picture disk, documentation)

Reviewed by: Heidi Imhof, Computer and Gifted Coordinator and Advisor to the Writer's Club, Connell Elementary School, North Franklin School District, Connell, WA

Story Maker is a text editing program for children, featuring the illustration capabilities of the KoalaPad, joystick or AppleMouse. Students may draw their own pictures or choose predesigned pictures from a gallery of three picture disks: a general gallery disk containing animals, trains, etc.; and two specialized disks—one on science fiction and one on fantasy. Eight enlarged font styles allow students to select and change the typeface.

The program is driven from a simple menu. In making a new story students use the KoalaPad, joystick or AppleMouse to select representative icons to type, select from the picture gallery, draw, erase, flip or clear pages, or exit. The same selection procedure is used throughout the program, simplifying instructions to the user.

Once a picture is selected, it may be "stamped" anywhere on a page. Students decide upon the size of their graphic (two sizes are available) and may flip each graphic 180 degrees. An UNDO function allows students to erase the last graphic stamped. For other changes there is an ERASE function that operates like an electronic pencil eraser.



Story Maker.

I had some initial questions when I began working with this program: These concerns included:

- Use of a single disk drive;
- Younger students' ability to use the KoalaPad when making menu selections;
- Lack of flexibility in the number of words that can fit on a single "page;"
- A few gallery pictures and one type style that appeared hazy and difficult to recognize on the screen;
- High cost of outfitting each writing section with a KoalaPad, AppleMouse or joystick;
- The tedious job of designating the printer with each printing.

After using this program with our Writer's Club (grades four through six), nearly all my concerns were laid to rest.

It should be mentioned that Scholastic provides teachers with lots of assistance in using this program with students. In addition to thorough documentation, the school edition includes ideas for introducing *Story Maker* in your classroom, as well as a whole section of teaching activities. This activity section has some standard writing ideas—good ones you have probably seen before—but these ideas are well adapted for use with *Story Maker*. Scholastic is exemplary in providing activities which allow their software to be introduced smoothly and integrated soundly into the curriculum. Here's what I discovered in using this program with students.

First of all, *Story Maker* is highly motivating. After showing students some of the things I had produced, we began writing a story together. Students couldn't wait for their turn to write stories of their own. After I trained a sixth grader on *Story Maker*, he made appointments before school, during lunch and after school to train the many students who were interested. Believing I had hit a fluke, I shared this program with groups of first through third graders with the same result. My first graders pored over the picture gallery cards (provided by Scholastic) and started their own stories while I worked one-on-one with each student at the computer. The KoalaPad section that I had worried about provided a good fine-motor, hand-eye experience for them, and they rose to the challenge, largely refusing my help. The letter and page size proved just right for primary students, who can become frustrated easily with the small print typical of many word processors. They made very careful and individual decisions about layout, and each story distinctly reflected its author. One first grader tried several locations, sizes and directions before she was finally satisfied with her string of horses "running in a pasture."

The problem of disk swapping was a concern only to some of my fifth and sixth graders who are proficient on *Bank Street Writer* already. My sixth grade trainer said, "It was worth having to change disks to get the pictures and backgrounds on *Story Maker*." Other older students thought they might use *Story Maker* for covers and illustrated pages and *Bank Street Writer* for the text of their stories.

The picture and type style that appeared hazy to me didn't trouble my students. After using the eighth type style once or twice, most students just opted to use one of the other, more legible, type styles. Many students took advantage of the program's ability to switch

fonts to add emphasis and interest to their stories. Pictures that were hard for me to discern were also no problem for my students. Although those brave few who decided to draw their own pictures learned how much mastery it takes to create a picture with the KoalaPad, experienced KoalaPad users were able to draw some great pictures.

In summary, my initial concerns were not valid for first through at least fifth grade. Experienced fifth and sixth graders generally recognized the tradeoffs and limitations, but were making sound decisions about appropriate use. However, I would still prefer using two disk drives, and would like to have *Story Maker* store the printer specification information.

Story Maker is simple to use, yet it takes good advantage of the computer's power. The ability to illustrate and write, with full layout control, gives children's creativity ample room for unique expression. Children can write and rewrite, as with most word processors, but can additionally redesign until they are content with their product. Scholastic has given us a fine word processing/storymaking program. After using it with children, I can heartily recommend it.

BANK STREET STORY BOOK

Mindscape, Inc., 3444 Dundee, Northbrook, IL 60062
Grade/Target Age Group: Grades 3-9
Subject: Creative writing and drawing
Hardware Required: Apple II+, IIe or IIc with at least 64K of memory, one disk drive, monitor or TV (preferably color), joystick or KoalaPad
Cost: $39.95
Reviewed by: Lynne Anderson-Inman, Ph.D., College of Education, University of Oregon

Bank Street Story Book is a new addition to the recent outpouring of commercial software designed to encourage creative writing by combining text processing with computer graphics. In a field that has captured the imaginative and productive interests of so many software developers, one might easily assume that the resulting products are roughly equivalent and that no single contender will stand out as truly exceptional. After several weeks spent examining and comparing some of these new programs, I would like to suggest that *Bank Street Story Book* is definitely something special.

Like other programs of its genre, *Bank Street Story Book* allows the user to create a multi-page story which contains both pictures and words. The story can be saved on disk and then revised, read or printed out whenever the user desires. What makes this particular program special is the range of artistic creativity and competence the program is capable of supporting, the sophistication of the word processing component, and the overall flexibility for mixing graphics and text in interesting and novel ways.

To create a picture, the user manipulates a "rubber band" cursor controlled through a joystick or KoalaPad. Upon selecting LINE from a vertical bar menu at the side of the screen, the user is free to draw anything anywhere. Use of the joystick's buttons allows the cursor to be lifted from the graphic in progress and set down somewhere else on the screen.

Bank Street Story Book.

Although the width of the line is constant, it can be drawn in any of eight different colors. For smaller detail work, and greater control, the cursor span can be decreased in size. Pressing "G" (for "Grid") constrains the cursor to move only horizontally or vertically, especially helpful for younger users who enjoy creating pictures from basic geometric shapes. Sections of a drawing can be filled with any of 21 colors. To accomplish this the user selects COLOR on the horizontal bar menu and then uses arrow keys to "mix" two of the available eight colors. A press of the joystick button and the desired color instantly fills the space targeted by the cursor. To make changes or corrections in a drawing, the delete key can be used to sequentially erase unwanted lines and colors.

To the novice joystick user, all of the above is somewhat foreign and awkward. Not having done anything of significance with a joystick before, I fumbled and fussed at great length before anything recognizable began to emerge. However, those familiar with the idiosyncrasies of this input device—even young children—seem to have no trouble adapting their already-acquired expertise to the specific demands of this program. When *Bank Street Story Book* was shared with some local second and third graders, those with previous joystick experience grasped the essential elements within a few minutes. The pictures are visually interesting and often quite stunning, especially those created by older users. Two sixth graders who experimented with the program were able to create some astoundingly sophisticated pictures; a sample story provided by the publishers indicates that a professional artist can go further yet. It's a credit to the designer that the program is easy enough for 7-year-olds, yet rich enough for adults.

To create a "storybook," however, the user needs more than an accessible graphics program. Without the ability to input text, there would be no story and hence relatively little academic value to the program. Unfortunately, this is where many such programs fall short. We still see products, for example, which accept only capital letters, do not accept the delete key, or use nonstandard procedures for altering text. *Bank Street Story Book* has avoided all of these traps. The word processing component on the program is sophisticated and easy to use. It has adopted the standard conventions for word processing programs (e.g., automatic word wrap, use of arrows to move cursor, easy insertions and deletions, etc.) and is even programmed to retain original text in memory, thus enabling revisions to be easily discarded if they are not felt to be improvements. The word processing component is accessible at any time in the process of creating a story by merely pressing "T" (for TYPE). A window appears which is then moved into place and expanded to the desired size with the joystick. A second pressing of "T" places the cursor at the top of the window, where it is ready to accept text.

From an instructional standpoint, the use of conventional word processing procedures in *Bank Street Story Book* is very desirable. Young students and those new to word processing may gain valuable skills through use of the program—skills that will later be transferable to other word processing applications. The corollary is equally true:

Students with previous word processing experience find the program easy to learn because it uses skills they already have. As expressed by one sixth grader, "Heck, this is a snap!" The typeface used in the program is relatively small (by comparison to other programs), and this is an advantage. It allows the writer to put quite a bit of text on a single page if desired, and it facilitates using the text for labels or story dialogue.

As powerful and interesting as these two components of *Bank Street Story Book* are, it is their integration which truly makes this program exceptional. Initially the user will probably work on a picture and then write the text to go with it, or write the text and then draw the corresponding picture. It is possible, however, to go back and forth from graphic to text as often as the user wishes, i.e., drawing a portion of the picture, putting in some text, altering or adding to the picture, adding or deleting some text, and so forth. Unless given the command to FLIP, all of this will be on one "page" of the storybook being developed. When read, each page appears on the screen in the order created (i.e., the graphic is drawn before your eyes, spaces are filled with color, and the text appears in the sequence it was entered). When creatively designed, the end result is a story page which appears to come alive. With some judicious use of the WAIT command (requiring the reader to press a button for the next portion of the page to be revealed) and some well-planned deletions, the user is even able to create simple animation and the appearance of dialogue between story characters.

At any point in this process, the user can ask to read through the story being created or choose to move back and forth one page at a time and one story step at a time. This allows for later changes in the pictures and editing of the text. The tremendous flexibility built into *Bank Street Story Book* has not resulted in a program which is difficult to use. It's a program that students can be successful with fairly quickly and then, as their experience builds, learn its finer features and capitalize on its depth. (Even second graders were fairly independent with the program by the second day and able to teach the essentials to their peers by the third day.)

The stories produced with *Bank Street Story Book*, when shown on the screen, are unlike anything else students have ever been able to create. The writer not only controls the content but also, to some extent, the rhythm and pacing with which the story is read. Pages become dynamic entities which both captivate and motivate. Instead of using a computer to create replicas of traditional static pages, *Bank Street Story Book* moves the writer and the reader into a new electronic literary medium.

The potential classroom applications for *Bank Street Story Book* are many and varied, the most obvious being a tool to facilitate creative writing instruction. When using the program with young students, it was apparent that the drawing process served to stimulate their imaginations, especially when drawing curved lines (i.e., not in the "Grid" mode). After doodling for awhile, their drawings began to look like something identifiable and, when prompted, they could write a few sentences to go with the picture. For teachers using the process approach to writing instruction, the program would work beautifully to generate ideas during the prewriting stage, while at the same time supporting the development and revision of students' written products. It is the program's capacity for accepting changes and facilitating corrections that seemed to intrigue the two sixth graders who tried it. Working as a team, they spent an en-

tire afternoon creating a story and then perpetually revising ("improving") it. Some of these improvements resulted from experimentation with the program's graphic capabilities, but a surprising percentage were text-based modifications and additions. In fact, the original story had approximately doubled in size by the time they felt it was complete.

Other possible applications might build on the program's ability to print out students' creations. Teachers using the language experience approach to reading instruction could type in dictated stories to support students' drawings and then print the individualized storybooks for later rereading and fluency building. Other students could use the program to create an illustrated classroom newsletter or a project report, complete with graphs, tables and maps.

It is always dangerous to write a review which doesn't appear to have taken a critical look at the program under examination. The enthusiastic and glowing praise of the above paragraphs may give the reader the impression that this is exactly what happened here. When beginning this review, however, I did have several areas of concern. First, the program's complexity made me wonder about its utility in the classroom: Would a teacher have time to learn and to teach students how to use it? Could the program be used by pairs of students or small groups? The first question was answered by the documentation. It is well constructed and well written. A tutorial is included to help teach the basic operations of the program quickly. Furthermore, on screen help menus are easily accessible and another tutorial is included on the disk. The second question was answered by observations of my sixth grade testers. They obviously enjoyed cooperating on the creative process and learned more about the program because of their joint efforts. Integrating *Bank Street Story Book* into the classroom should pose no major problem to teachers already familiar with microcomputers.

A second concern centered on the apparent inappropriateness of the program for children in the primary grades, even though the publishers had targeted children as young as eight. After all, if *I* had some trouble manipulating the joystick and learning the program's idiosyncrasies, they would surely feel overwhelmed! Thanks to cooperative local teachers, some last-minute testing was done with second and third graders. Observation of these young users revealed they not only loved the program, but that they could learn it easily—even from their peers.

Only my third concern remains unresolved. I would like to have at least one simple, sample story on the disk that novices can identify with and younger children can read. I felt frustrated when my drawings were in no way comparable to those included as samples (even the samples created by children!). This concern may be more a reflection of my artistic talents (or lack thereof) than the program's limitations, but it is something to consider—especially when introducing the program to younger children.

"The stories produced with BANK STREET STORY BOOK ... become dynamic entities which both captivate and motivate. BANK STREET STORY BOOK moves the writer and the reader into a new electronic literary medium."

Bank Street Story Book is clearly a program to explore and I recommend it very highly. It is one of those rare programs which continues to unfold as you use it. For teachers who value the role microcomputers can play in facilitating student writing, and who are looking for a motivating way to introduce or encourage the use of this new medium, *Bank Street Story Book* will undoubtedly become a useful and highly effective classroom tool.

[The reviewer would like to thank Larry Lewin of Eugene School District 4J for his assistance in testing the program with second and third graders.]

Reviews

Linda Rathje, Editor

Story Tree
Graphics Exhibitor
South Dakota
Dancing Bear

Teacher, Kids and Logo

A comparative review of 3 computer dictionaries

STORY TREE

Author: George Brackett
Scholastic, Inc., 906 Sylvan Avenue, P.O. Box 2010, Englewood Cliffs, NJ 07632
Target Age Group: Ages nine and up
Subject: Writing
Hardware Required: Apple IIc, Apple IIe, Apple II Plus (48K), or Apple II (with 64K or Applesoft in ROM); IBM PC, IBM PCjr (128K); Commodore 64.
One disk drive, monitor or TV, printer optional.
Cost: Retail edition, $39.95, includes Master Disk, Story Disk and User's Handbook.
Teacher's edition, $59.95, includes 2 Master Disks, 2 Story Disks, User's Handbook and Teacher's Guide.
Reviewed by: Lynne Anderson-Inman, Ph.D., College of Education, University of Oregon, Eugene, OR

Everyone loves a good adventure story, especially when the reader plays a role in deciding how the adventure will unfold. George Brackett's *Story Tree* capitalizes on this interest and takes it one step further. What could be better than ***reading*** "choose-your-own-adventure" stories? Well, ***writing*** them, of course! *Story Tree* lets you and your students do just that.

Story Tree is an exceptionally well designed piece of software that facilitates both the writing and reading of "interactive" stories, i.e., stories in which the reader makes choices determining the direction of events. Although an obvious application for a writing tool of this nature is in the creation of adventure stories, *Story Tree* lends itself to the development of other interactive genres as well. Provided as samples on the Story Disk are a somewhat silly jungle adventure in search of the magical Marigold Mine, a well researched and interesting article on the creature known as Bigfoot, and a mini data base of extremely brief book reviews organized by topic. Each uses the program's capabilities in a slightly different way and together they help to illustrate some of its applications.

The *Story Tree* program takes its name from the branching capabilities of the stories that can be created with it. When writing with *Story Tree*, the writer has three options for linking pages together:

a) A story can CONTINUE from one page to the next, as it would in any story;

b) The reader can be given a CHOICE from among two to four options, with each option leading to a different series of events or type of information; and

c) The story can be made to branch by CHANCE, leading in one direction some of the time and in another for the remainder.

If used creatively and in combination, the resulting stories can be read and reread without being repeated in exactly the same sequence.

Designed for children as young as nine, *Story Tree* is a program that has been constructed with care. The main menu provides the user with four choices: (a) Read a story, (b) Work on a story, (c) Print a story and (d) Do disk work. Each choice leads to an appropriate submenu or a listing of the available stories, and ESCAPE always brings you back to the main menu. When reading a story, the user can elect to read it from the beginning or s/he can choose another place to start by browsing through the page menu. The latter is especially useful when rereading stories, as it allows curious readers immediate access to different events and alternate endings. It is also useful when editing stories, as the page-by-page access to different parts of a story helps the writer to locate pages needing clarification or story lines needing development.

When writing a story, the user is prompted through three operations for each page: (a) writing a page title (to allow access through the page menu); (b) writing text for the story; and (c) indicating how the page should be linked to successive pages, i.e., will it CONTINUE to another page automatically, branch by CHOICE or branch by CHANCE? After making a selection on the latter, the user is asked a sequence of questions which enables the program to create the appropriate connections between and among pages. For example, if branching by CHOICE, what choices are provided to the reader and to which page should each choice lead? Although the above may sound complicated, doing it is actually fairly simple, in large part because of the excellent prompts presented to the user on screen. For writers who haven't thought out all the details of a story (e.g., only one of three alternative choices has actually been conceptualized), temporary choice descriptions and page designations (letters or numbers, for example) can be used and then changed to something more descriptive when details have crystallized. As they are being created, all *Story Tree* stories are saved on a story disk, allowing for later reading or editing. Stories can be altered or expanded at any time, thus encouraging authors toward greater detail and increasingly complicated plots.

Story Tree stories can also be printed at any time. Two *Story Tree* pages are printed on an 8½ x 11 piece of paper, and each section is marked with a page number. When cut and appropriately arranged, small books can be made from the printed pages. The capacity to print out stories is useful to students in at least two ways. First, it allows stories to be shared with family and friends. Second, having a printed copy enables the stories to be read and reread when the student is not near a computer, thus extending the educational value of the program and promoting the transfer of learned skills to noncomputerized settings. The reinforcing nature of the printed stories would be enhanced, however, if an optional large print mode were available. It

BEST COPY AVAILABLE

is important for young students, as well as for older students with reading and writing problems, to see a big payoff for such a big investment of effort.

Suggestions for Use

In preparing this review I used *Story Tree* with two students (ages nine and 14), and I talked with teachers who had used the program with small groups of students or entire classes. The following suggestions for using *Story Tree* represent a blending of ideas from these experiences and discussions.

1. Prior to introducing *Story Tree*, it seems best to ensure that students have had experience with "choose-your-own-adventure" stories.* Teachers found that students who had had previous exposure to this type of story in book form approached *Story Tree* enthusiastically. Those who had not had this exposure were often unclear as to why a story should be read more than once, and were more confused about the structure of *Story Tree* stories, especially when writing them. If grouping students for work on *Story Tree*, take into consideration their experience with this type of writing. Students with little or no experience with "choose-your-own-adventure" stories should be grouped together, otherwise they tend to be overwhelmed by their more enthusiastic and confident peers and usually end up as passive participants in the creative process.
2. Collect, swap and exchange *Story Tree* stories created by students. One of the best ways to be introduced to *Story Tree* is for students to read sample stories, either in pairs or as a small group. Although sample stories are provided on the disk, they are not appropriate for all levels of students and are not written by children. The 14-year-old I shared the program with felt the stories were too juvenile, and went on to create a complex James Bond-type spy story. The nine-year old felt Magic Marigold Mine was funny but complicated, and she couldn't identify with the report on Bigfoot at all. One of the best features of *Story Tree* is that it will support a wide variety of story types on an unlimited number of topics. Developing a library of stories for students to explore prior to writing their own will help to show off the program's possibilities and broaden its use.
3. Before writing *Story Tree* stories, it is especially helpful to teach students to map out or flowchart the sequence of events or information. This can be introduced using experiences from students' lives (to help convey the idea of choice and the impact of choices made on future events) and then by translating these experiences into the terminology used in the *Story Tree* program. The manual provides an excellent tutorial on how to write *Story Tree* stories and uses a mapping technique to illustrate the different types of page connections. Most teachers found this helpful in teaching students how to plan and keep track of their stories. (See figure 1 for the map used in this tutorial.) In general, those students who took the time to map out their stories were able to finish them; those who did not often ended up in confusion.
4. Integrate the use of *Story Tree* with practice on word-processing skills. Built into the program is an excellent and easy-to-use word processor. It can be used to introduce students to the principles of word processing or as an alternate arena in which to apply already learned word-processing skills. Although most of the word-processing commands in *Story Tree* are fairly standard, teachers will need to pay special attention to those that differ from word-processing programs students are already using.
5. If using *Story Tree* in a classroom setting, provide students with a five-minute warning before the end of the period. There is no way a

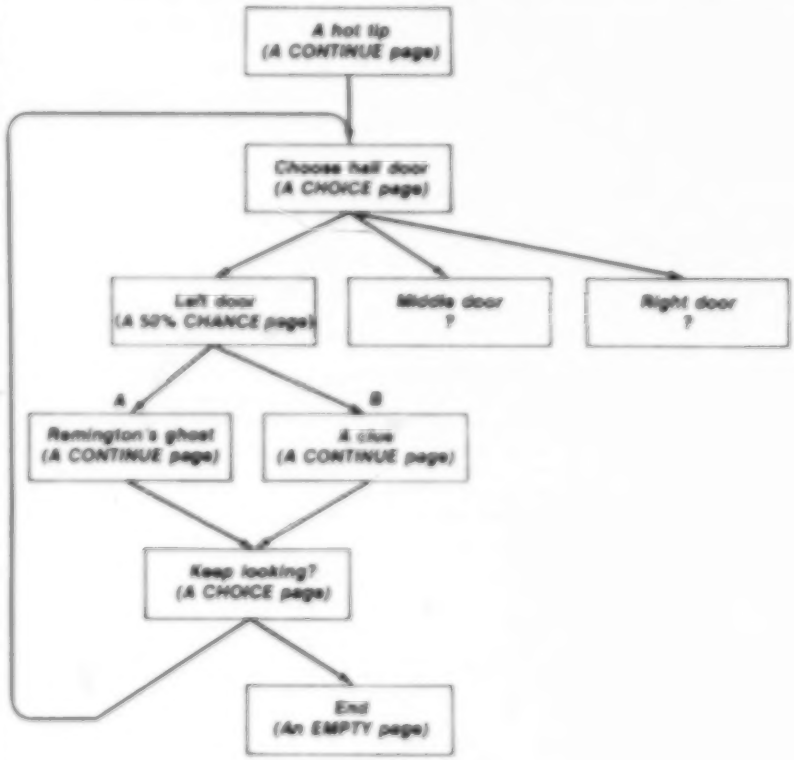


Figure 1.
mapping the sequence of events when writing Story Tree stories.

*Several book publishers produce books of this nature with varying levels of reading difficulty: *Choose-Your-Own-Adventure* by Bantam Books, *Twistaplot* by Scholastic Books, and *Which Way?* by Archway Books are three series that can usually be found in bookstores and in school or public libraries.

page that is being worked on can be saved before completion. This means that in order to save any text that has been written, the student must also designate the type of page connection and the destinations for each option before being able to stop. This requires some time—hence the need for a five-minute (or longer) warning.

Other Applications

Story Tree is not limited to the language arts or English classroom. And even in these classrooms, the program is not limited to the development of stories. The Teacher's Manual which accompanies the educational version of *Story Tree* does an excellent job in suggesting other uses for the program and even includes guidelines for creating special types of "shells"—frameworks of pages linked together for specific purposes. A sampling of these other applications is enough to highlight the underlying potential of *Story Tree*.

In the area of social studies, for example, groups of students could use the program to write an imaginary tour of another country or another era in time. Information could be organized around specific topics, as well as by location or date, allowing readers a different experience each time they took the "tour." In science, students might write a description of the circulatory system as seen from the "inside." Different branches of the descriptive report would address different branches of the system. A class poem could be generated using the random poetry shell, a framework that takes advantage of the branch by chance option. Each student writes a rhyming couplet on the same theme. When entered into the shell, the poem can be read and reread many different times with many different results. As a final example, the program can also be used to generate on-screen multiple-choice learning activities. Choosing the correct option leads to a "right answer" page and then on to another question. All incorrect choices lead to explanatory pages with individualized corrective feedback and then a return to the original question. It is suggested that students might develop these learning activities as well as play them. Although I was unable to find teachers who had implemented any of these ideas, or others from the manual, the ideas appear to be instructionally sound.

Making a Good Program Even Better

I've yet to see a software program that couldn't be improved upon in some way, and *Story Tree* is no exception. What is already a very good program can be made even better. The following suggestions for improvement come from teachers and students who have used the program extensively and therefore build on an in-depth appreciation for everything *Story Tree* already provides.

1. It would be helpful if *Story Tree* would automatically map or flowchart a story as it is being created. Younger writers, and those with special learning problems, often become "lost" in their stories, not being able to sort out which pages went with which story lines. Being able to consult a map of the story in progress would help to minimize this confusion and concentrate student energies on the writing process.

2. Students should have the option of seeing page titles on the printed version of their stories. As currently printed, page numbers are substituted for page titles. Since the pages are identified by title and not by number on screen, this substitution makes it extremely difficult to cross reference the printed version with the on-screen version. When editing, students often use the printed version to keep track of undeveloped story lines or to identify pages needing modification. Inclusion of page titles on printed stories would facilitate this aspect of the editing process.

3. It would be educationally helpful if *Story Tree* stories would allow use of the commonly available spelling checker programs. Teachers have become increasingly convinced of their value in promoting good spelling habits, and many teachers now encourage the use of such programs on all student writing. There is no logical reason why *Story Tree* stories should be an exception.

4. And finally, students should be directed to develop title pages for their stories. Although there is nothing to prevent the creation of title pages for *Story Tree* stories, there is no mention of title pages in the documentation and no examples provided on the disk. In the eyes of my nine-year old tester, "real" stories have title pages. Providing directions for their creation would be an easy and valuable addition to the program.

Overall Recommendation

In summary, I recommend Scholastic's *Story Tree* very highly! It is a well designed program that is educationally sound and fun for students to use. Furthermore, it is extremely versatile. It could find a comfortable home at many different grade levels, and teachers will discover interesting applications in a number of content areas. Because of its flexibility, it is appropriate for classrooms with a wide range in student ability and can be used quite successfully with special needs students. It is appropriate for students to use independently, but equally effective if used with students in pairs or in small groups.

Of equal importance to teachers should be the fact that students like *Story Tree*, and once the basics are learned will often spend their own time embellishing stories begun in class. This seems to be true even for populations of students traditionally turned off to writing. One of the teachers interviewed for this review related that he had several ninth grade boys come into his computer lab daily, on their own time, in order to write with *Story Tree*. These boys taught themselves the program and then spent approximately an hour a day, five days a week, for six to eight weeks writing and rewriting their stories. Any writing software that elicits that kind of commitment deserves to be looked at seriously!

Acknowledgement

The reviewer would like to thank the teachers of Eugene School District 4J who shared their students and their experiences using *Story Tree*. Special thanks go to Tom Layton, computer teacher at South Eugene High School, who graciously shared ideas arising from his use of *Story Tree* with students who have special learning needs and also provided many of the suggestions for improving the program.

Publisher's Reply: None.

Doris Carey

The Bank Street Writer
Personal Use: © 1982 from Broderbund Software, Inc., 1938 Fourth St., San Rafael, CA 94901. Cost: $62.50.
School Use: © 1983 from Scholastic, Inc., 902 Sylvan Ave., Englewood Cliffs, NJ 07632. Cost: $95.00.
Grades 4-12/Word Processing.
Hardware Description: Apple II, Apple IIe, Atari. (Commodore 64 being developed.)
Reviewed by Keith Wetzel, University of Oregon, Eugene, Oregon.

The *Bank Street Writer* (BSW) is a word processor designed for students. It represents a good use of the computer in an upper elementary or junior high school classroom.

Typically word processors have been adult-oriented. They require days to learn to operate, thus they are not accessible to students. In contrast, the BSW requires only one to two hours to learn to use. No knowledge of programming or math is necessary to use it.

The BSW potentially enables a classroom teacher to use the ideas of process writing—prewriting, writing and rewriting—efficiently in the classroom. One of the keys to improving student writing is to encourage students to write more. A word processor appropriate for students is motivating, and reduces the drudgery associated with editing and recopying.

How It Works

The BSW is organized like a three-page menu. The first page is the writing or text-entering page. The user begins here. The words ENTER TEXT are always highlighted at the top of the page. The first page allows the word processor to be used much as a typewriter. The move to the EDIT page is made by pressing the ESC (escape) key, with the words WRITE OR CORRECT highlighted at the top of the screen on this page.

The philosophy of the BSW's authors is evident. They wanted one mode for entering text. After the user completes entering all his/her ideas, then it is time to move to another distinct mode to correct errors, edit sentences or move ideas around.

The editing page has this menu at the top:

ERASE MOVE FIND TRANSFER
UNERASE MOVEBACK REPLACE

The user chooses the correct command by using the forward and backward arrows. When the correct choice is highlighted, the user presses the return key and follows the directions which appear at the top of the screen. The commands do what their names suggest. The user does not need to remember them because they are always present at the top of the screen.

The third page of the menu is the TRANSFER text page. This page allows the user to save the typed text. Then the text can be printed on paper, deleted from memory or retrieved from memory. Once again the choices are listed at the top of the screen.

There are error traps to prevent mistakes. The user is asked, "Are you sure you want to . . .?"

Apple II vs. Apple IIe

This word processor has upper and lower case letter capability on both Apples. The Apple II+ requires the user to press two keys (the shift key and the N key) to print a capital letter. A minor modification to the Apple II+ enables the user to type a capital letter using only the shift keys. The Apple IIe requires no modification to type capital letters in the usual format. The IIe also has a shift-lock feature which enables the user to type continuous capital letters.

The BSW can be used on either Apple. The same piece of software will operate on both machines. The Apple II+ uses the I J K M keys to move the cursor in the text edit mode. The Apple IIe has separate arrow keys ←↑↓→ to accomplish the same task. The BSW works appropriately with either machine, and the tutorial also reflects these differences.

Printed Copy

The BSW uses any standard printer and requires no special set-up. To print copies the user is required to follow the directions after highlighting TRANSFER and then PRINT FINAL. A series of questions appear which allow the user to adjust the number of characters printed per line, the spacing (single, double or triple), the numbering of each page and other printing details. After each copy the user will be asked if another copy is desired.

Limitations

There are few limitations of the BSW for intermediate and junior high age students. One limita-

tion might be the difference between what appears on the screen and what appears on the printed copy. The BSW allows forty characters to be printed per line on the screen. The number of characters printed per line are adjusted prior to printing. The standard typing format is about sixty-five characters per line. Consequently, the printed text does not appear in identical form to the text on the screen. This does not seem to cause students any problems.

Broderbund and Scholastic

There are two versions of the BSW—they differ primarily in the audience they are directed towards, and consequently the way they are packaged and marketed. Broderbund distributes the program only to individuals, their market aimed primarily at home users. There is no preview or review period prior to purchase. It includes two disks and a small reference guide at a cost of $62.50. Scholastic distributes the program only to schools. It includes three disks and a teacher's manual. The tutorial side of the disk may be copied if you wish to use the tutorial with groups. Schools are allowed a thirty-day preview period. The cost is $95.00.

Documentation or Teacher's Guide

Broderbund's user manual is concise. In combination with the tutorial, which is on the opposite side of the disk, it answers questions and gives the user practice in entering and editing text.

The Scholastic version shares the same software. Though the word processor is the same and the tutorial is the same, the documentation is different, as it is designed to meet the needs of teachers and students. It includes a teacher reference guide, a student reference guide, a question and answer section and a student activities section. The student reference material can be copied and distributed to students.

The BSW is not easy for students in intermediate grades to learn to use—guidance is necessary for quick, efficient learning. I wanted to find out how easily a seventh grader would learn to use the BSW with only the assistance of the reference material. In forty-five minutes, and with a few questions, this student learned to use it.

How does that compare with another seventh grader using only the tutorial? The tutorial beeped often, indicating a student error. When the student made an error, the tutorial did not offer hints or an analysis of the error. The consistent beeps and the message, "You did not follow directions," frustrated the student. The student's comments upon completion of the tutorial in fifty minutes was, "It was tedious, but I did learn something." The student who used only the reference material and asked a few questions had a more positive experience.

Several intermediate grade level teachers also commented that the tutorial was not appropriate for their students. Their reasons were similar to the problems discussed earlier.

The BSW may not be understandable to the student user at first, but a judicious use of the reference material, the student activities, a teacher's oral introduction and the tutorial would seem to be an effective way to allow students to use it independently after a few hours of practice.

Rating

Both versions—Broderbund and Scholastic—contain an excellent word processor which meets a real need. The Broderbund streamlined package is fine for home use. And the Scholastic package with an additional disk and a teacher's manual is appropriate for school use. Why is the BSW excellent? First, it is an educational piece of software which does what it was intended to do—make an advanced writing tool accessible to parents, students and teachers. Second, it compares favorably in price to other commercial word processors. Third, the students who I questioned about rewriting a composition by hand or editing it using the word processor emphatically endorsed the word processor. A fifth/sixth grade teacher who uses BSW consistently for language arts activities summed it up: "I like it, but the kids *love* it."

Broderbund Distributors:

UK: Tele Swift, 189 London Rd., North End, Portsmouth, England.

Australia: Imagineering, P.O. Box 4601, Sydney, Australia, NSW 2001.

Reviews

This month TCT's reviews include two well-known products that have been expanded or revised. With *Apple Logo II*, the larger memory and special features of the 128K Apple IIe and IIc are available for Logo programmers.

The *Bank Street Writer*, on the market since 1982, has recently been revised. There are now several versions of this popular word processor available through both Scholastic and Broderbund. Also reviewed this month, the *Bank Street Speller* provides several features that may be of particular interest to both students and teachers.

For primary-grade teachers new to Logo, a resource kit by Houghton Mifflin is available. *Introducing Logo to Primary Children* provides a classroom kit of teacher-developed instructional support materials.

Two French foreign-language programs are also featured. *Les Batisseurs d'Empire*, the French-language version of *Empire I: World Builders*, has been available in the U.S. for over a year. After examining several programs, this science fiction simulation game was selected by the reviewer as one of her favorites for third and fourth year French students. *Le Francais Par Ordinateur* offers three beginning French programs. Color graphics, music and a variety of learning activities introduce the student to French vocabulary and the culture.

BANK STREET WRITER

1984, Revised Apple Version
Personal use edition:
Broderbund Software, Inc., 17 Paul Drive, San Rafael, CA 94903
School use edition:
Scholastic Inc., 902 Sylvan Ave., Englewood Cliffs, NJ 07632.
(Includes three program disks, Student Guide, Teacher's Reference Guide)
Target Grades: 4-8
Subject: Word processing
Hardware Required:
Broderbund—
40-column version: Apple II+ (64K); Apple IIe (64K)
40/80-column version: Apple IIe (128K) Revision B board; Apple IIc
1 disk drive
Scholastic—
40-column version: Apple IIe (64K)
40/80-column version: Apple IIc (128K); Apple IIc
1 disk drive
Cost: $69.95 (Broderbund); $95 (Scholastic)
Reviewed by: Keith Wetzel, Computers in Education, University of Oregon

When the first *Bank Street Writer (BSW)* came out, many of us were caught by a vision of the promise it held for writing in the elementary and middle schools. Since that time I have closely observed students using word processors in the intermediate grades, and my euphoria has been tempered by classroom realities. Accordingly, this review concentrates on the new *BSW*, but it also reflects new knowledge about the use of word processors by students in grades four through eight.

As a writing teacher, my primary criterion for any word processor is that it allows students to move easily and efficiently between the different stages of the writing process. Writing is not a linear, lock-step process of pre-writing, composing and revising. Rather the process is recursive: The writer moves back and forth between pre-writing, composing and revising. This process continues until the writer is satisfied with the product.

The new versions of *BSW* for the Apple IIe and IIc, marketed by Scholastic and Broderbund, are substantially different from the original *BSW*. Each now has a free-moving cursor which may be moved around the screen in the WRITE mode to make insertions and deletions. ***This feature makes the new version significantly better than the original version.*** In addition, it satisfies the evaluation criterion, allowing students to move easily and efficiently between the stages of writing.

The original *BSW* and the updated Broderbund version when used on the Apple II+ (64K) hamper revision by adopting a two-mode system: one mode for entering text and another for editing. For example, seven different steps are required to jump up two paragraphs, add a new sentence and return to continue writing. It is much easier for a student to use a word processor which allows text to be edited as it is written. For this reason, *BSW* in any edition for the Apple II+ is clearly less desirable.

Apple IIe (64K) Features

BSW for the Apple IIe (128K) and IIc can do everything the IIe (64K) version can do, and more. The Apple IIe (64K) version has these new features:

- Full cursor movement on the write screen
- Printer format commands (boldface, underline, etc.)
- Optional mouse control to select functions or scroll text
- Help screen
- End-of-line character
- New-page command

Apple IIe (128K) and IIc Features

Additional features available only on the Apple IIe (128K) and IIc versions are:

- Choice of anywhere from 40- to 80-column display
- Calculator for solving basic math

Features Missing

The *BSW* developers have included some new features to make this word processing program more powerful, and other features have been left out. One missing feature is that the screen display does not necessarily show the printed format; that is, what you see is *not necessarily* what you get. With some word processing programs what you see on the screen is exactly what is printed onto paper.

Missing also is a straightforward way of single spacing portions of text in a double-spaced report.

Documentation

The documentation for the Broderbund versions and the teacher's guide for the Scholastic version are clear and helpful.

Discussion

Developers of the new *BSW* have had to make decisions concerning what new features to include. Questions needing to be addressed include: "What do students in grades four through eight need in order to write more efficiently?" "What features are needed to allow *BSW* to keep up with the features of other word processors?"

The first question is clearly answered. The decision to allow free cursor movement in the WRITE mode certainly makes the new *BSW* a more powerful tool for writing. The other question must be answered individually by the purchaser, on the basis of those features needed for his/her purposes.

Composition Teachers

Teachers who want students to use any word processor (including *BSW*) in learning composition need to fully understand what word processors actually do. Recently a fourth-grade teacher told me, "I still have to teach my entire writing program. What is really important is what the teacher *teaches* about writing. It is not simply the computer." This reflects an important point: Word processors do not teach writing. They do not tutor students or substitute for a writing program. The word processor, rather, is a tool in the writing process.

My second caution concerns expectations: It is commonly assumed that students will write more if they can revise without recopying by hand. Before jumping to this conclusion, consider carefully whether your students can type adequately. Without this prerequisite skill, students may not write as much with a word processor as they do with paper and pencil.

Recommendations

My review of the original *BSW* appeared in the August 1983 *TCT* (Vol. 11, No. 1), when it was the only word processor designed for fourth through eighth graders that I knew of. Now there are other word processors designed specifically for students, as well as adult word processors which may be considered for students because they are not difficult to learn or use.

It was also easier for me to be enthusiastic before realizing the importance of adequate keyboarding skills and acquiring a thorough knowledge of the teaching of composition. Now word processors have been in classrooms long enough that we can "look before we leap."

Consequently, I advise that you carefully examine the idea of using word processors in composition, and if you decide to take the leap, use several word processors yourself before making a decision. Despite my cautionary tone, I find the new *BSW* is one of the top word processors available for students in grades four through eight, and I recommend you include it on your list of word processors to be examined.

Publisher's Reply

Thanks very much for sending me the review copy for the new *Bank Street Writer*. Generally, I found the review to be fair and accurate. But I did want to point out a couple of things:

The reviewer mentions that "the screen display does not necessarily show the printed format." While this is technically true, it is possible to get, quite quickly and easily, printed copies that mirror what the screen shows—and vice versa. If you are using the 40-column version, selecting "Print-Draft" automatically gives you a printout that uses the same line length as the document shown on the screen. If you are using the 80-column version, an option in the Utility Program allows you to set the screen display for 40 columns, 80 columns, or any line width between 40 and 80 columns. As a result, once you decide on the line width that you want to use in your printed copy, you can set the screen for the same display.

The Scholastic school edition of the new *Bank Street Writer* includes, on the same disk, versions that will work with the 64K Apple IIe, the 128K IIe and the IIc. This is an important feature for schools that have a mix of 64K and 128K IIe's, for schools that currently have IIe's and are planning to upgrade them with an extended 80-column card, and schools with IIe's that plan to purchase IIc's.

The reviewer also remarks that schools will have to determine for themselves which of the other enhanced features meet their needs. While this is very true, I think we've done a good job of developing and testing features that make the *Bank Street Writer* an even better product for both students and teachers. As we mention in the documentation, while the new *BSW* is just as easy to use as the original version, the expanded edition allows students to grow with the program, gradually uncovering new features as they become comfortable with the more basic functions.

Philip Miller
Senior Project Editor
Scholastic Software

BANK STREET SPELLER (1984)

Personal use edition:
Broderbund Software, Inc., 17 Paul Drive, San Rafael, CA 94903
School use edition:
Scholastic, Inc., 902 Sylvan Ave., Englewood Cliffs, NJ 07632
Subject: Spelling checker
Hardware and Software Required: 48K Apple II+/IIe/IIc
One disk drive (two required for adding words to the dictionary)
Bank Street Writer word processor
Cost: $69.95 (Broderbund); $64.95 (Scholastic)
Reviewed by: Doris Carey, Computers in Education, University of Oregon

Word processor users have definite ideas about what spelling checkers should do. They should, of course, point out spelling errors, and in addition should refrain from correcting these errors until instructed to do so, in case the word has been deliberately misspelled or it is a proper noun or foreign word that does not appear in the standard English dictionary. *Bank Street Speller* accomplishes these objectives by comparing each word in the user's text file to its 30,000-word dictionary (derived from the *Random House Dictionary*, Concise Edition) and allowing the user the opportunity to decide whether to replace the word or leave it intact. *Bank Street Speller* can't guarantee the correct spelling of words that are not in its dictionary, nor can it point out errors in style or grammar—but neither can a regular dictionary.

Bank Street Speller provides several additional features that may be of particular interest to teachers: an alphabetic list of words used with frequencies; the total count of words used; and a count of *different* words. My small sample file, for example, used a total of 52 words, but only 45 different words, since "with" was used twice and the words "a," "the" and "to" were each used three times. The list of different words with their frequencies can be printed out or viewed on screen. The totals appear only on the screen. With a bit of imagination, a teacher can use this feature for games and activities that enhance students' awareness of their vocabularies. Students might be asked to draw a histogram that illustrates their use of two-letter, three-letter, four-letter, etc., words. They might also make a game of writing a coherent paragraph in which no word is used more than once. The user can also quickly alphabetize a list of words.

Bank Street Speller will also print out a list of words that it did not find in its dictionary. Again, students can make a game of picking out words that are probably misspelled, as opposed to exotic words or proper nouns that should be retained. Teachers can decide what to do about slang, since words like "ain't" do not appear in the dictionary. Words categorized as "obscene" in my *American Heritage Dictionary* also seem to have been left out of *Bank Street Speller*. The **Make New Dictionary**

feature allows the user to add or delete words to the dictionary and to create new dictionaries. I see great possibilities here for foreign language teachers.

Bank Street Speller's **Search Dictionary** feature is particularly interesting. It prints a list of words in the dictionary that *almost* match a particular misspelled word. For example, I included "woud" in my text file, and a search of the dictionary yielded the words "wood," "would," "word" and "wound" as suggested replacements. I was also able to check the spelling of "separate" by entering "sep?rate." Entering "r = d" gave me a list of all words in the dictionary that begin with "r" and end with "d." The question mark allows for a single character, whereas the equals sign is a wild card that allows for the substitution of any number of characters.

After I finished making changes to my text file, I decided to retain 14 of the 18 words that *Bank Street Speller* had not found in its dictionary. The four misspelled words that I did replace were printed out for me in this form:

woud ⟶ would

As a teacher, I was happy to see that students could keep a list of these corrections in a scrapbook or notebook for future reference or review.

Bank Street Speller is easy to use. Like its parent word processor, *Bank Street Writer*, the program's menus are always visible at the top of the screen, with questions and prompts appearing on screen when needed. As an experienced user of *Bank Street Writer*, I was able to fully check, edit and print out my file in 10 minutes the first time I attempted to use *Speller*. This included the time it took to obtain printouts of my word lists, errors and corrections. I spent another minute or two accessing the utility section to customize the program to accept a keyboard modification for upper and lower case letters (since I was using an Apple II+).

Teachers who plan to purchase this program should be aware that single-drive systems will require a vast number of disk changes. Once *Speller* was loaded, I had to load my file from my data disk, then alternate between the dictionary and data disk each time I needed to search the dictionary. Without these searches, I needed at least five disk changes from beginning to end, including reloading *Bank Street Writer* and the data file to print out the final copy. I would hastily add that these annoyances are not due to a design flaw, but to the limits of the technology. It would be desirable to have both programs in memory simultaneously, but the obvious tradeoff would be a requirement of more than 48K of available memory.

One small design decision that I would change is the arrangement of word lists. They are alphabetized, but appear in two columns arranged to be read left to right; I would prefer to scan down a column instead of across rows.

I would strongly recommend the purchase of *Bank Street Speller*, not only because it fulfills its objectives, but because its many features provide an environment for experimentation and creative play. I expect it won't be long before an enterprising teacher compiles a list of activities for use with this program—I wish I had the time to do it myself.

BEST COPY AVAILABLE

QUILL

DCH Educational Software, D.C. Heath and Company, 125 Spring Street, Lexington, MA 02173
Target Grade Group: 3-12
Subject: Writing
Hardware Required: Apple II + (with 64K) or IIe, two disk drives, printer, 80-column card, green screen monitor (for greater clarity)
Cost: $150 (school price)
Reviewed by: Larry Lewin, Fourth Grade Teacher, Adams Elementary School, Eugene, OR

As a writing teacher at the elementary, secondary and college levels for 10 years, I have come to respect any writing program that understands the four-step approach to writing. When I was told that D.C. Heath had marketed *Quill*, a software program developed under a U.S. Department of Education grant, I was anxious to try it out. *Quill* is more than a word processor; it incorporates the four steps of the writing process: prewriting, composing, revising and publishing. I expected *Quill* to assist young writers in working through each step in the process to improve their writing, and anticipated a computer program surpassing both the traditional textbook instructional method and other computer software available. I now question *Quill*'s ability to do that.

Program Overview

Quill consists of three sections: the Planner, which assists in the pre-writing phase; the Library, which serves as a storage place for sharing written work; and the Mailbag, an electronic mail simulator designed to encourage students to communicate with each other by passing electronic notes. Each section is stored on a separate initialized disk and accessed when loaded into a required second disk drive. The program's word processor, the Writer's Assistant, is used to revise compositions stored in the Library.

To familiarize students with each component, the Teacher's Guide provides an introductory section, called the Cookbook, for teaching *Quill*. It also includes a set of classroom activities, an extended section teaching each command of the Writer's Assistant in an easy-to-follow sequence, and a command card. Since I had some difficulty in remembering the command steps, I referred to the Teacher's Guide frequently.

"I have four concerns with QUILL: disk access time, the text editor, spacing options and the Planner options."

Strengths

Several features of the Library impress me. First, as I typed a draft of this review, I accidentally lost two superbly written paragraphs. I pushed CTRL C in a vain attempt to call up the deleted text, but *Quill* eventually let me retrieve it by escaping through Quit, waiting out the 45-second respite, and answering the question, "Which would you like to do?" I hastily selected "4) Forget the changes just typed and leave the entry as it was." A happy reunion.

In the Library menu, a nice safety device has been included. Choosing See allows a student to read a classmate's story without unauthorized tampering. They can discuss the story, and suggestions can be made. If the author does decide to change the story, s/he selects Change to edit.

I like the menu display across the top of the screen, and the mnemonic menu items (D for Drop, M for Mix, Q for Quit, etc.). Left and right margins are adjustable, so newspaper columns, for example, can be formatted. Automatic paragraph indentation and line skipping between paragraphs present a useful format for young writers. Files can be labeled by title/author (or co-authors if you assign pairs), or by file number (chronologically). Even though these labeling options slow the exit from the Write mode, I believe teachers will like having them.

The most exciting feature of *Quill* is its ability to Mix a paragraph into a list of sentences, each beginning at the left margin. The young author can then analyze each sentence for structure, length, capital letter in first word, punctuation, and paragraph compatibility. Two quick keystrokes Mix the sentences back into paragraph form. This feature is an outstanding use of the technology to improve writing (see figures 1 and 2).

Keywords: /Review/

larry seahorse

A sea horse lives in the sea. but it is not really a horse. It is a sea animal that has a head like a horse. It is very cute. Its tail is long and curly. The tail is used to hang onto plants while it eats.

Figure 1.

Figure 2.

Keywords: /Review/

larry seahorse

A sea horse lives in the sea. but it is not really a horse.
It is a sea animal that has a head like a horse.
It is very cute.
Its tail is long and curly.
The tail is used to hang onto plants while it eats.

Another component, Mailbag, allows the student to send and receive messages to and from other individuals identified by code, or to groups via the Bulletin Board. Feedback on a classmate's story can be transmitted in this way. The Teacher's Guide claims Mailbag enhances writing instruction by encouraging various types of writing, offering varying audiences for writing, and motivating writing by personalizing it. Mailbag does all three.

© Copyright 1985 ICCE

Concerns

I have four concerns with *Quill*: disk access time, the text editor, spacing options and the Planner options.

It takes almost two minutes to boot the disk, move through title pages, and make menu selections before arriving in the Write mode. Just to Quit and move out of Write can take from 45 seconds to over a minute, depending on the Quit menu selection. If I wanted to See an entry, and then decided to work on it some more, I had to leave See mode, return to the menu, select Change, and head back to the entry: one minute and 35 seconds. The fourth grade students I observed tended to engage in off-task behaviors, in spite of the phrase written across the monitor, "Watch those disk drives spin along, doo-dah, doo-dah." *Quill* does offer a shortcut allowing the user to move directly between Planner, Library and Mailbag, but at a mere *six-second* savings.

The text editor, Writer's Assistant, is the weakest component of *Quill*. A word processor must allow students to move quickly between the stages of revision. Writing is not linear; it is not a lock-step process of pre-writing, composing and revising. Rather, the process is recursive, moving from composing to brainstorming to editing and back to composing in any order until the writer is satisfied. In my opinion the *Quill* editor hampers revision by adopting a multi-mode system. The Write mode, which both enters and edits text, is subdivided into two separate commands: In and Drop. In allows the author to draft new text and to insert. Drop deletes letters or whole words. Since the author cannot delete and insert within the same command mode, s/he must move back and forth between these two modes, a process requiring seven keystrokes. It is much easier to use a text editor which allows text to be edited as it is written.

The word processing programs that are most straightforward are those which automatically adjust the format of a paragraph every time text is inserted or deleted. In contrast, the *Quill* editor is line oriented. When new text is inserted, the line is pushed into the right margin. When the margin buffer is full, the remainder of the text is temporarily erased from the screen. Although text reappears when CTRL C is pressed, there is no automatic reformatting of the paragraph. Reformatting requires yet another multi-keystroke procedure.

I was surprised to find that double or triple spacing options are not available when typing in text or when setting printer commands. The lack of space between lines makes editing of hardcopy drafts frustrating and crowds the appearance of the final copy.

The Planner component of *Quill* offers some pre-writing assistance, but is no more than an electronic worksheet. The teacher types an introductory sentence and then lists questions for the student to consider. Student answers serve as notes for the writing assignment. It is pre-writing, but does not use the technology to its potential. I respect *Quill*'s inclusion of this component of the writing process, but it lacks branching capability or the use of graphics. And, there is no option for transferring a completed Planner to the Library.

Recommendation

Quill is marketed as a tool for the "writing process approach to language arts." The Teacher's Guide discusses the four stages of that process, and the software offers a component for each stage. But I was disappointed. The Planner is limited in its ability to assist young writers in the pre-writing stage. It does no more than a worksheet, listing leading questions. The Writer's Assistant is a word processor with below-average text-editing capabilities for the drafting and revising stages.

Quill is not quickly learned. It took me many hours of practice with the manual open to the command section before I felt competent using it. And the slowness of the program in booting and moving between the sections of each program compares unfavorably with other word processors available for students.

I recommend that teachers thoroughly examine *Quill* and compare it with other writing programs before purchasing. To their credit, D.C. Heath has included a 30-day approval policy providing teachers an opportunity to determine whether *Quill* meets the writing needs of their students.

Publisher's Reply

DCH Educational Software appreciates very much the opportunity to respond to your review. While we do not share all the reviewer's perceptions of *Quill*, we have chosen not to present an item-by-item defense of the program. Rather, we join the reviewer in urging teachers to examine *Quill* for themselves. In addition, we recommend that they consult the reviews of *Quill* that have appeared in the following publications:

Teaching and Computers (November/December 1984)
Classroom Computer Learning (March 1985)
Popular Computing (January 1984)
The English Journal (March 1985)
Computers, Reading, and Language Arts (Spring 1984)
The Digest of Software Reviews (January 1985)
Early Years K-8 (March 1985)
School Tech News (September 1984)

During the more than two years that *Quill* has been in classroom use, it has been our great pleasure to hear from scores of satisfied teachers who have found *Quill* to be an effective tool for teaching the writing process. One, writing from Alaska, told us that "kids took to *Quill* like a sled dog to pulling." *The English Journal* says, "If you want a program that will motivate students to enjoy writing, help them organize their pre-writing, make revisions easier, and make publication simple, then *Quill* is what you are looking for."

The best way to evaluate a software program, however, is not to read about it, but to use it. DCH's 30-day preview policy makes it easy for teachers to see for themselves whether *Quill* is indeed what they are looking for.

Jim Harmon
Software Marketing Coordinator
DCH Educational Software

© Copyright 1985 ICCE